



UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE TECNOLOGIA

Missão/FT: Formar e aperfeiçoar cidadãos e prestar serviços atendendo às necessidades tecnológicas da sociedade com agilidade, dinâmica e qualidade.



Guilherme Mazoni

CONTRIBUIÇÕES PARA O SENSORIAMENTO E RASTREAMENTO DE ATLETAS VIA ZIGBEE

Limeira, 2015

Guilherme Mazoni

Contribuições para o Sensoriamento e Rastreamento de Atletas via ZigBee

Contributions for sensing and tracking athletes by ZigBee

Dissertação apresentada ao Curso de Mestrado da Faculdade de Tecnologia da Universidade Estadual de Campinas, como requisito para a obtenção do título de Mestre em Tecnologia.

Área de Concentração: Tecnologia e Inovação
Linha de Pesquisa: Engenharia da Informação

Orientador: Prof. Dr. Rangel Arthur

ESTE EXEMPLAR CORRESPONDE À REDAÇÃO
FINAL DA DISSERTAÇÃO DE MESTRADO DO
ALUNO GUILHERME MAZONI.

Limeira, 2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Felipe de Souza Bueno - CRB 8/8577

M458c Mazoni, Guilherme, 1992-
Contribuições para o sensoriamento e rastreamento de atletas via Zigbee /
Guilherme Mazoni. – Limeira, SP : [s.n.], 2015.

Orientador: Rangel Arthur.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de
Tecnologia.

1. Sensoriamento remoto. 2. Esportes. 3. Atletas. 4. Sistema de
Posicionamento Global. I. Arthur, Rangel, 1977-. II. Universidade Estadual de
Campinas. Faculdade de Tecnologia. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Contributions for sensing and tracking athletes by Zigbee

Palavras-chave em inglês:

Remote sensing

Sports

Athletes

Global Positioning System

Área de concentração: Tecnologia e Inovação

Titulação: Mestre em Tecnologia

Banca examinadora:

Rangel Arthur [Orientador]

Paulo Roberto Pereira Santiago

Mauro Menzori

Data de defesa: 14-01-2015


Programa de Pós-Graduação: Tecnologia

DISSERTAÇÃO DE MESTRADO EM TECNOLOGIA
ÁREA DE CONCENTRAÇÃO: TECNOLOGIA E INOVAÇÃO


Contribuições para o Sensoriamento e Rastreamento de Atletas via ZigBee

Guilherme Mazoni

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. Rangel Arthur
Faculdade de Tecnologia (FT)
Presidente



Prof. Dr. Paulo Roberto Pereira Santiago
USP/Ribeirão Preto-SP



Prof. Dr. Mauro Menzori
Faculdade de Tecnologia (FT)

Resumo

Este trabalho apresenta uma solução para o rastreamento de atletas em tempo real, denominado Sistema de Sensoriamento e Rastreamento (SSR), com a finalidade de auxiliar na análise e no desempenho de atletas. Por prover resultados em tempo real trata-se de uma solução inovadora de avaliação de desempenho dos atletas. Neste trabalho, busca-se contribuir no rastreamento em duas dimensões utilizando a tecnologia de Global Positioning System (GPS) e a técnica de trilateração, por meio do indicador *Received Signal Strength Indicator* (RSSI) disponibilizado pelo padrão de comunicação sem fio IEEE 802.15.4, popularmente conhecido como ZigBee®. A integração das tecnologias envolvidas neste trabalho provê aplicação em diversas modalidades esportivas, pois tem a possibilidade de operação em locais fechados, abertos e diversos tipos de ambientes. O SSR em conjunto com a tecnologia ZigBee® permite o sensoriamento dos atletas em tempo real, fornecendo dados complementares para as análises de desempenho, provenientes, por exemplo, de sensores analógicos e digitais, acelerômetro, giroscópio e magnetômetro. Para isso é proposta uma plataforma com microcontrolador de baixo custo. As simulações e resultados obtidos provam a eficiência do sistema proposto e permite especificar os dispositivos mais interessantes a partir de diferentes ambientes de práticas esportivas.

Palavras-chave: rastreamento, sensoriamento, esportes, atletas, GPS, IEEE 802.15.4.

Abstract

This work presents a solution for tracking athletes in real time, called System Sensing and Tracking (SSR), in order to assist in the analysis and performance of athletes. By providing real-time results it is an innovative solution performance evaluation of athletes. This work aimed to contribute to the trace in two dimensions using the Global Positioning System technology (GPS) and trilateration technique, with the indicator Received Signal Strength Indicator (RSSI) provided by wireless communication standard IEEE 802.15.4, popularly known as ZigBee®. The integration of technologies involved in this work provides application in several sports, it has the

possibility of operating indoors, open and wide range of environments. The SSR together with ZigBee® technology allows the sensing of real-time athletes, providing additional data for performance reviews, from, for example, analog and digital sensors, accelerometer, gyroscope and magnetometer. For this we propose a platform with low-cost microcontroller. Simulations and results prove the efficiency of the proposed system and lets you specify the most interesting devices from different environments of sportive practices.

Keywords: tracking, sensing, sports, athletes, GPS, IEEE 802.15.4.

Sumário

Resumo	vii
Abstract.....	vii
Sumário.....	ix
Agradecimentos	xi
1. Introdução	1
1.1 Histórico.....	1
1.2 Motivação	2
1.3 Objetivo do Trabalho	2
1.4 Contribuições	3
1.5 Estrutura do Trabalho	3
2. Revisão Teórica	4
2.1 Rastreamento de Objetos	4
2.2 Exatidão e Precisão	4
2.2.1 Analogia com o tiro ao alvo	4
2.3 Tecnologia IEEE 802.15.4/ZigBee®	5
2.4 Estimação da posição por valor de RSSI	6
2.5 Método matemático para estimação da posição por RSSI.....	7
2.6 Métodos matemáticos para o cálculo da Trilateração.....	8
2.7 Tipos de GPS	10
2.7.1 GPS por Código.....	10
2.7.2 GPS por fase	11
2.8 Trabalhos relacionados com a tecnologia ZigBee em rastreamento de objetos	11
3. Materiais e Métodos	13

4.	Composição do sistema	16
4.1	Dispositivo Móvel.....	16
4.2	Estações fixas para Trilateração	19
4.3	Propriedades do módulo GPS.....	20
4.4	MPU-9250	21
4.5	Propriedades físicas do módulo ZigBee.	22
4.6	Considerações sobre a estrutura física do sistema	24
4.7	Interface de sensores no sistema.....	24
4.8	Topologia de rede do sistema	25
5.	Testes e Resultados.....	26
5.1	Experimento de análise do padrão de erro em posicionamento absoluto por código (C/A) 26	
5.1.1	Metodologia.....	26
5.1.2	Resultados.....	28
5.2	Experimento comparativo de curvas de RSSI aferidas e teórica.	31
5.2.1	Metodologia.....	31
5.2.2	Resultados.....	33
5.3	Experiência com o método de Trilateração com o MATLAB.....	35
5.4	Experiência de coleta de dados GPS do sistema.....	40
5.4.1	Metodologia.....	40
5.4.2	Resultados.....	46
6.	Conclusões.....	56
6.1	Trabalhos Futuros	57
7.	Bibliografia.....	58
8.	Anexo e Apêndice	60

Agradecimentos

Agradeço a Deus pelo dom da vida a mim concedido de forma graciosa por todos os dias que já vieram e aos que hão de vir até o fim deste momento.

À minha família, meu presente, que me motiva a seguir em frente sempre apoiando as minhas decisões com amor e carinho.

Aos meus pais, meus heróis.

À minha namorada, por todos os sorrisos sinceros, pessoa na qual adoro compartilhar meus momentos.

Ao meu orientador Prof. Dr. Rangel Arthur, por participar na minha formação como cidadão e por todas as oportunidades oferecidas que inspiraram a criação deste trabalho e com certeza muitos ainda.

Aos Prof. Dr. Milton Shoiti Misuta da FCA e ao Prof. Dr. Mauro Menzori da FT, no envolvimento e toda a ajuda cedida na realização deste trabalho.

Ao meu amigo Rodrigo L. Ximenes que por agradável convivência me ensinou muito e que me apresentou a fabulosa área dos microcontroladores.

Aos meus irmãos da República Ross Galaxy, por todos os momentos de risadas que não me deixaram desanimar.

Por fim, agradeço a todos os meus amigos e irmãos de fé, principalmente aos quais me dão confiança e motivação para seguir em frente com humildade e respeito ao próximo.

1. Introdução

Um sistema de rastreamento em tempo real captura a posição de objetos e ou pessoas a disponibilizar séries de coordenadas espaciais medidas em um curto período de tempo. Essas séries de coordenadas podem ser utilizadas em navegação e em outras aplicações que dependem do posicionamento para a formação de propriedades estatísticas.

1.1 Histórico

Atualmente é notado o crescimento e popularização de muitas modalidades esportivas. Como consequência, surgem novas ferramentas e tecnologias para ajudar os atletas no desenvolvimento e monitoramento de suas práticas, sendo disponíveis bases estatísticas para o acompanhamento do atleta. A posição do atleta durante a prática esportiva é uma informação que depende do seu rastreamento e permitem aos respectivos treinadores a realização de análises complexas de desempenho.

Este trabalho consiste na utilização do GPS e uma ferramenta da tecnologia ZigBee, o RSSI, a fim de compor um sistema de rastreamento e sensoriamento de atletas de forma abrangente, com aplicação em ambientes fechados e abertos.

Os receptores GPS atuais funcionam basicamente por um dos dois modos, código e fase, sendo o de código o mais acessível e utilizado em aplicações civis. As variações de tipo envolvem principalmente a precisão e a exatidão, com custo proporcional a estas características.

As aplicações de um receptor de GPS por código são consideradas de baixa precisão e exatidão se comparadas ao receptor GPS de fase. Um exemplo desse caso ocorre durante a navegação de um veículo terrestre, que sempre é feita acompanhada em um mapa como base referencial, caso haja discrepâncias muito grandes elas serão desconsideradas pelo usuário. Neste trabalho, o uso de receptor GPS por código, em aplicações de rastreamento em tempo real de precisão e exatidão, é comparado com o receptor GPS que usa a fase.

O uso do GPS de fase deve ser considerado no sistema de rastreamento de atletas, porém devido ao seu custo elevado, um modelo simples, por exemplo, tem cerca de 10 vezes o custo de modelo de código (C/A), sendo menos acessível ao projeto.

É fato conhecido que a tecnologia GPS não atua eficientemente em ambientes fechados, pois está submetida a interferências e problemas de visada direta dos receptores com os satélites. Assim, em anos recentes, [1] as redes de sensores sem fio (WSNs) estão sendo estudadas para a localização de objetos em ambientes fechados. O ZigBee é uma tecnologia que pode compor uma WSN e, desse modo, pode ser utilizada para o rastreamento de atletas em ambientes fechados.

1.2 Motivação

Atualmente, os dispositivos comerciais possuem diversas limitações no rastreamento de atletas. Além de terem custo elevado não oferecem aos usuários os dados que muitas vezes não possuem a confiabilidade e a precisão necessárias para análises mais complexas. Por meio do sistema composto, em parceria com o grupo do Laboratório de Biomecânica da Faculdade de Ciências Aplicadas da UNICAMP, tem-se como finalidade, a obtenção satisfatória de sinais sensoriais de atletas possibilitando análises que atualmente não são possíveis por dispositivos comerciais.

1.3 Objetivo do Trabalho

Este trabalho tem como finalidade a composição de um sistema de baixo custo, o SSR, com a tecnologia IEEE 802.15.4/ZigBee e o Sistema de Posicionamento Global (GPS) para o rastreamento e sensoriamento de atletas em diversas modalidades esportivas. Isso foi possível a partir de técnicas de trilateração e estimação de distâncias por intensidade de sinal de radiofrequência, com aplicação em tempo real e com o auxílio de sistemas embarcados em ambientes fechados e por meio de GPS, para ambientes abertos. Assim, o SSR disponibiliza uma solução para o rastreamento de atletas em tempo real e também permite a interface de sensores

em atletas, com o propósito de facilitar e inovar a forma de análise da prática de atividades dos atletas, em diversas modalidades esportivas.

1.4 Contribuições

Dada às características do trabalho, o mesmo contribui para:

- Solução de rastreamento de objetos em geral.
- Proposta de sensoriamento de rede sem fio, composição de uma *Wireless Sensor Network* (WSN).
- Aumento na competitividade de produtos comerciais de rastreamento aplicados a esportes.
- Aperfeiçoamento assistido do desenvolvimento de atletas.
- Possibilidade de surgimento de técnicas inovadoras na análise esportiva.
- Confecção de patentes.

1.5 Estrutura do Trabalho

O trabalho é dividido em seis capítulos, sendo na primeira seção é apresentada uma introdução do trabalho com a apresentação de um histórico, motivação e objetivo, além das possíveis contribuições.

O Capítulo 2 apresenta uma revisão teórica sobre as tecnologias empregadas e de trabalhos relacionados. Materiais e métodos estão descritos no Capítulo 3 e, logo em seguida, a composição do sistema no Capítulo 4.

Os testes e resultados dos experimentos realizados com o sistema são exibidos no Capítulo 5, finalizando com as conclusões no Capítulo 6.

2. Revisão Teórica

Neste capítulo serão definidos importantes conceitos relacionados às técnicas de rastreamento tradicionais e as tecnologias que podem ser usadas para rastreamento. Além disso, alguns termos são definidos para que se possa entender sobre a eficiência dessas técnicas.

2.1 Rastreamento de Objetos

RTK, do inglês, cinética em tempo real, tem como fundamento a captura do movimento de objetos em uma frequência de aquisição muito alta, oferecendo exatidão em nível de centímetros e correções de posição em tempo real dada uma posição de referência. [2]

Particularmente, o RTK é um método de posicionamento que usa receptores GPS que adquirem as fases das portadoras, onde é necessário ao menos um dispositivo fixo em uma posição conhecida e um dispositivo móvel onde se comunicam por um enlace de rádio. O RTK na determinação das coordenadas utiliza uma técnica que também pode ser empregada em outros tipos de sistemas de posicionamento.

2.2 Exatidão e Precisão

Exatidão e precisão são dois termos distintos, mas que são comumente confundidos ou até mesmo considerados iguais pelos mais leigos. Uma das maneiras de se avaliar a qualidade do resultado de uma medição é fornecida pelo conceito de exatidão, que se refere à proximidade da medida com seu valor verdadeiro. Outra qualidade muito importante de uma medida experimental é seu grau de precisão, que se refere à dispersão entre medidas repetidas sob as mesmas condições. [3]

2.2.1 Analogia com o tiro ao alvo

Para compreender melhor os conceitos de exatidão e precisão, é usual se fazer analogia entre o processo de medição e um exercício de tiro ao alvo. Na base dessa analogia está a ideia de

que, assim como o objetivo de um atirador é atingir o centro do alvo, o objetivo da medição é determinar o valor verdadeiro da medida. A Figura 2.1 ilustra quatro resultados possíveis em um teste de tiro.

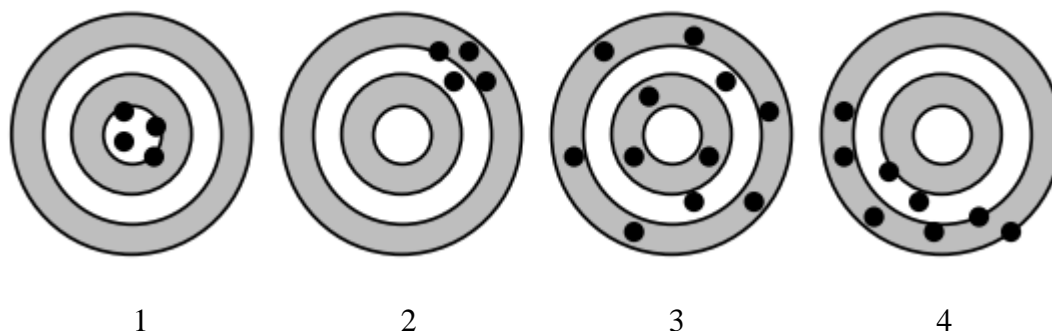


Figura 2.1 - Quatro resultados de tiro ao alvo. [3]

Nos quatro casos da Figura 2.1, os tiros estão dispostos de diferentes maneiras. Nos casos 1 e 2, eles estão menos dispersos que nos casos 3 e 4. Também é possível perceber que, nos casos 1 e 3, os tiros estão distribuídos em torno do centro enquanto, nos casos 2 e 4, a distribuição de tiros está descentralizada.

Na nossa analogia, isso quer dizer que, nos casos 1 e 2, há mais precisão. Ou seja, quando a medição foi repetida sob as mesmas condições, ela produziu resultados semelhantes (menos dispersos). Nos casos 1 e 3, há mais exatidão porque as medidas estão distribuídas em torno do centro do alvo (o valor verdadeiro). [3]

2.3 Tecnologia IEEE 802.15.4/ZigBee®

Um módulo de RF ZigBee é uma solução embarcada que disponibiliza conectividade sem fio entre dispositivos. Os módulos utilizam o protocolo de rede Digimesh para conexões ponto-a-ponto em frequência de operação livre de 2.4 GHz. Uma rede ZigBee pode transferir dados em até 250kb/s em distâncias de até 300 metros em ambientes fechados. O módulo ZigBee também tem baixo consumo de energia comparado as tecnologias de comunicação competidoras de

topologia similar, como Bluetooth e Wi-Fi. O ZigBee, Bluetooth e WiFi quando utilizados em aplicações móveis, possuem, respectivamente, duração de bateria de anos, dias e horas. [4]

Uma característica interessante e muito importante para a aplicação neste trabalho com a tecnologia ZigBee é a disponibilidade da Intensidade de Sinal de Rádio Recebido (RSSI), que pode, em conjunto com a teoria de propagação de sinais eletromagnéticos, determinar a distância entre os nós da WSN, consequentemente disponibilizar as coordenadas de um objeto acoplado ao módulo ZigBee.

2.4 Estimação da posição por valor de RSSI

O requisito básico para a determinação de coordenadas espaciais de um nó em 2D em uma rede WSN ZigBee é a existência de três estações fixas de referência, que são responsáveis por aferir os valores de RSSI com a estação móvel. Estes três valores são convertidos em distâncias e então é feita a trilateração da estação móvel, assim, determinando a posição do objeto, no caso da aplicação, o atleta acoplado ao módulo ZigBee da estação móvel.

A aferição do valor RSSI de um módulo ZigBee é feita por comandos AT, através da interface serial (RS232 TTL/CMOS), estes comandos AT são provenientes de um microcontrolador PIC interligado à WSN ou a partir de um computador com um dispositivo adaptador de interface USB.

Também é possível aferir os valores de RSSI por um pino elétrico específico do módulo ZigBee (pino 6).

A estação coordenadora pode ser uma das três estações fixas de referência espacial ou uma estação isolada ao sistema de posicionamento, para o segundo caso, esta é conectada a algum tipo de interface humana, por exemplo, um computador ou um *tablet*.

Tendo obtido os valores de RSSI aferidos pelos módulos ZigBee, estes são convertidos em dBm, dado que, eles são disponibilizados pelos módulos em valores hexadecimais com referência ao último dado recebido de ponto da WSN, esta conversão pode ser feita na expressão do valor hexadecimal em uma variável do tipo decimal assinado, que agrega valores negativos, por exemplo, um valor de RSSI de -84 dBm é disponibilizado como 0xAC (Hexadecimal) que é equivalente a 172 (decimal não assinado).

Como os valores de RSSI adquiridos por comandos AT são sempre números inteiros é possível afirmar que a precisão da estimação fica limitada a resolução dos valores de RSSI. A presença de números de domínio real, de RSSI, durante a solução da posição pode ser resultado de operações estatísticas como a média aritmética.

Seguindo as características do fenômeno de propagação de ondas eletromagnéticas no espaço e tendo os valores de potência dos sinais em dBm é possível determinar a distância entre dois nós da rede WSN, pois a potência utilizada na transmissão é conhecida, isso é possível, dado que a energia de um sinal eletromagnético transmitido é atenuada exponencialmente ao longo da distância transmitida.

2.5 Método matemático para estimação da posição por RSSI.

Tendo como referência a Fórmula de Friis (2.1) que relaciona potência do sinal transmitido e recebido (P_t e P_r), ganhos das antenas dos transceptores (G_t e G_r), comprimento de onda da frequência de transmissão (λ) e a distância (d), aplicando as propriedades da rede ZigBee, pode se considerar os ganhos e comprimento de ondas como constantes, reduzindo à Fórmula (2.2) expressa em Watts, onde (k) é uma constante de calibração disponibilizada pelos fabricantes de módulos ZigBee de forma sugestiva e adaptável. Sendo (n) um fator de propagação, no caso, $n = 2$ para ambientes de visada direta e aplicando os valores em dBm do RSSI aferido chega-se a Fórmula (2.3) para o cálculo da distância 1D.

$$\frac{P_r}{P_t} = G_t \cdot G_r \left(\frac{\lambda}{4\pi d} \right)^2 \quad (2.1)$$

$$P_r = \frac{k}{d^n} \quad (2.2)$$

Logo,

$$RSSI = -10n \log(d) + k \quad (2.3)$$

Isolando d

$$d = e^{-\frac{1}{20}R - \frac{1}{20}k} \quad (2.4)$$

2.6 Métodos matemáticos para o cálculo da Trilateração.

Os métodos matemáticos a seguir foram propostos por [1] para a utilização das distâncias aferidas e composição dos valores de coordenadas estimados da estação móvel. Este é o processo de trilateração, vide Figura 2.2, onde é conhecida a posição (X_i e Y_i) de ao menos três estações de referência e as distâncias (d_p^i) entre a estação móvel e as de referência fixas.

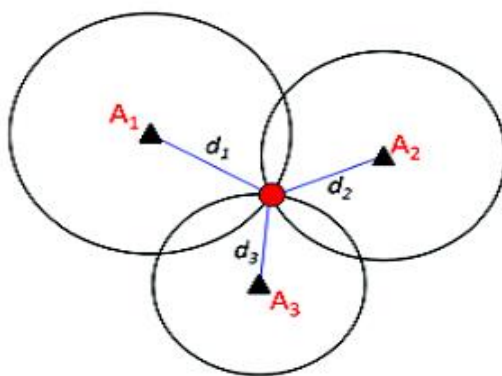


Figura 2.2 - Processo de trilateração, onde A_1 , A_2 e A_3 são as estações referência. [4]

A posição da estação móvel $P(X_p, Y_p)$ pode ser obtida por mínimo erro quadrático médio (MMSE) [5]. A diferença entre a distância atual e a estimada é definida pela Fórmula (2.4) onde (i) é uma referência de posição e (p) a estação móvel.

$$d_p^i = \sqrt{(X_i - X_p)^2 + (Y_i - Y_p)^2} \quad (2.5)$$

A Equação 2.4 pode ser expressa como:

$$(d_p^i)^2 = (X_i - X_p)^2 + (Y_i - Y_p)^2 \quad (2.6)$$

Então a considerando as três ou mais estações de referência a Equação 2.5 pode ser transformada em:

$$\begin{bmatrix} (d_p^1)^2 - (d_p^2)^2 + (X_2^2 + Y_2^2 - X_1^2 - Y_1^2) \\ (d_p^1)^2 - (d_p^3)^2 + (X_3^2 + Y_3^2 - X_1^2 - Y_1^2) \\ \dots \\ (d_p^1)^2 - (d_p^N)^2 + (X_N^2 + Y_N^2 - X_1^2 - Y_1^2) \end{bmatrix} = \begin{bmatrix} 2(X_2 - X_1) & 2(Y_2 - Y_1) \\ 2(X_3 - X_1) & 2(Y_3 - Y_1) \\ \dots & \dots \\ 2(X_N - X_1) & 2(Y_N - Y_1) \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \quad (2.7)$$

Onde,

$$b = \begin{bmatrix} (d_p^1)^2 - (d_p^2)^2 + (X_2^2 + Y_2^2 - X_1^2 - Y_1^2) \\ (d_p^1)^2 - (d_p^3)^2 + (X_3^2 + Y_3^2 - X_1^2 - Y_1^2) \\ \dots \\ (d_p^1)^2 - (d_p^N)^2 + (X_N^2 + Y_N^2 - X_1^2 - Y_1^2) \end{bmatrix}$$

E,

$$A = \begin{bmatrix} 2(X_2 - X_1) & 2(Y_2 - Y_1) \\ 2(X_3 - X_1) & 2(Y_3 - Y_1) \\ \dots & \dots \\ 2(X_N - X_1) & 2(Y_N - Y_1) \end{bmatrix}$$

Portanto, a Equação (2.6) é transformada na Equação (2.7), na qual pode ser solucionada utilizando a matriz dada pela Equação (2.8) na qual a posição $P(X_p, Y_p)$ pode ser obtida.

$$b = A \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = (A^T A)^{-1} * (A^T b) \quad (2.9)$$

2.7 Tipos de GPS

Atualmente existem dois tipos de funcionamento mais comuns para o GPS, estes são citados com mais detalhes a seguir.

2.7.1 GPS por Código

Normalmente, os receptores GPS geram uma cópia de uma sequência binária contida no sinal enviado pelos satélites e a utilizam para alinhar os sinais. Uma vez que o sinal transmitido pelo satélite leva algum tempo para alcançar o receptor, as duas sequências, inicialmente não coincidem, a cópia do sinal do satélite é retardada em relação à cópia local. Quanto mais a cópia local atrasa as duas cópias podem então ser eventualmente alinhadas. O atraso correto representa o tempo necessário para que o sinal alcance o receptor, a partir deste tempo, a distância entre o satélite e a base local pode ser calculada. [6]

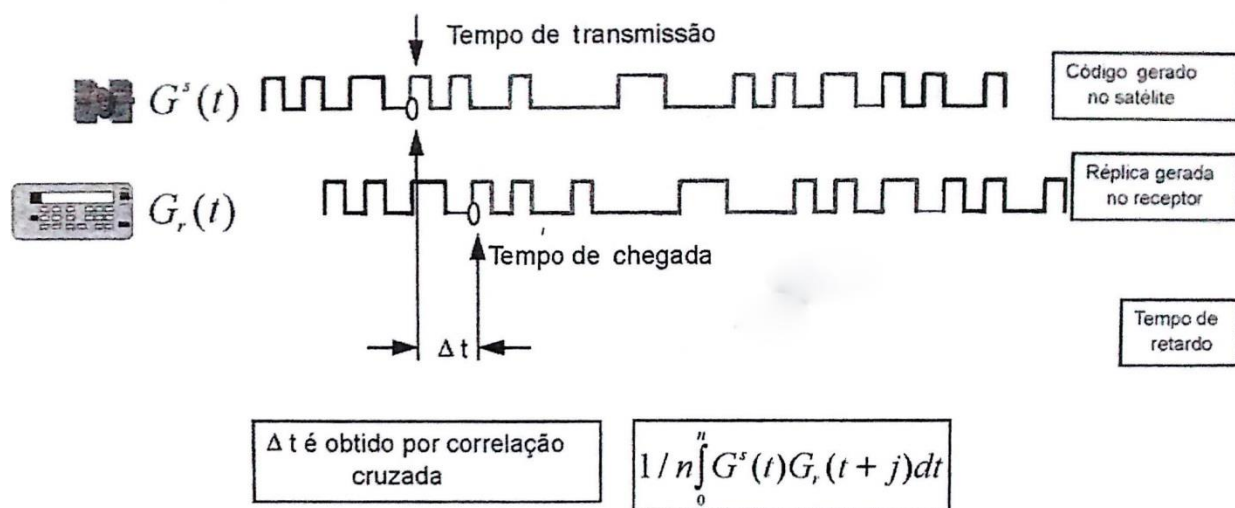


Figura 2.3 - Princípio da medida de pseudodistância. [7]

A precisão da medida resultante é essencialmente uma função da capacidade da eletrônica do receptor para processar com precisão os sinais do satélite. Em geral, os receptores são capazes

de alinhar os sinais em cerca de 1 % da largura de um bit. Por exemplo, o código de aquisição (C/A) enviado sobre o sistema GPS envia um bit a cada 0,98 microssegundos, então o receptor é preciso em 0,01 microssegundos, ou cerca de 3 metros. Outros efeitos introduzem erros muito maiores do que esta preposição, a precisão na prática, com base em um código (C/A) não corrigido é geralmente em torno de 15 metros. [6]

O funcionamento de código militar P(Y) é similar ao de código (C/A), porém ele é enviado com um intervalo de tempo dez vezes menor, ou seja, 0,098 microssegundos, consequentemente, precisão de cerca de 30 cm. Adotando os mesmos efeitos de erros no código (C/A) a precisão fica em torno de 1 metro. Porém o código militar (P) é criptografado (Y) para fins de segurança e indisponível para fins civis. [6]

2.7.2 GPS por fase

O GPS por fase segue o mesmo conceito geral, mas usa a fase da portadora da onda do sinal do satélite como sua referência, ignorando as informações de código para o cálculo da posição. A melhoria proveniente desse tipo de sinal é potencialmente muito elevada, se continuando a assumir uma precisão de 1% na recepção. No caso do GPS de código (C/A) a aquisição do código (*broadcast* na banda L1) muda de fase a 1,023 MHz, mas a portadora L1 em si é de 1575,42 MHz, mais que mil vezes a frequência do código. A frequência da portadora corresponde a um comprimento de onda de 19 cm, um erro de $\pm 1\%$ na medição da fase da portadora L1 corresponde a um erro de $\pm 1,9$ mm. Porém, para um GPS que usa a fase funcione, a necessidade eletrônica é mais cara. [6]

2.8 Trabalhos relacionados com a tecnologia ZigBee em rastreamento de objetos

A referência [8] mostra um algoritmo baseado em filtro de Kalman, onde se faz da predição e estimação do sinal, para a determinação da posição dos objetos em uma WSN e segundo [9] é possível ter-se alta precisão na determinação da distância, através de métodos que correlacionam o domínio do tempo e o da frequência em um algoritmo que não utiliza RSSI, mas a diferença de tempo de chegada (TDOA) dos dados da rede WSN.

Este trabalho utiliza a proposta do rastreamento em tempo real (RTT), sugerida pela referência [1], onde métodos mais simples para a determinação da distância são utilizados. A comparação desse trabalho com outros trabalhos mostra que os resultados aqui obtidos têm características de precisão, exatidão e repetitividade inferiores aos demais, porém aceitáveis para a aplicação de rastreamento de atletas.

3. Materiais e Métodos

Este trabalho envolve aplicações na área de esportes, assim, ao longo do desenvolvimento deste Mestrado, foram realizadas reuniões com professores da Faculdade de Educação Física e da Faculdade de Ciência Aplicadas da UNICAMP, que sugeriram características de funcionamento ideais para o SSR, com prioridade, precisão e integração dos sensores.

Para atender às necessidades e possibilitar o desenvolvimento do projeto, primeiramente foi necessário definir o microcontrolador a ser utilizado, ou seja, o “cérebro” do *hardware*. Uma vez que definido o microcontrolador, a metodologia se baseou no desenvolvimento de interfaces ao dispositivo, conseqüentemente os testes com o sistema pronto. O SSR envolve até dois meios de posicionamento, todos eles podem ser desenvolvidos separadamente, devido à característica de montagem modular do sistema, assim sendo integrados pelo microcontrolador. Esta vantagem garante que cada tipo de sensor utilizado funcione de forma independente e com máximo desempenho, agregando ao SSR as melhores características de cada um, como as do GPS, trilateração por Zigbee e possivelmente a integração de sensores inerciais do MPU-9250 [10].

Uma vez definido o microcontrolador, foi desenvolvida uma série de testes com cada tipo de sensor que poderia ser utilizado, onde cada sensor rendeu um estudo para seu melhor aproveitamento, permitindo o projeto da interface com o *hardware*. A série de testes deve caracterizar o sensor em precisão, exatidão e restrições de funcionamento.

Para o GPS foi necessário desenvolver algoritmos de transformação e leitura dos dados fornecidos pelos dispositivos GPS, aplicando técnicas de descrição de erros e testes em tempo real, para sua caracterização.

O sistema de trilateração com ZigBee requeria o desenvolvimento de conversores de modulação PWM de precisão para melhor confiabilidade dos dados de RSSI, também estudos com a topologia de rede. O Zigbee além de ser uma ferramenta de posicionamento em conjunto com técnicas de trilateração é o dispositivo que proporciona a comunicação sem fio dos demais sensores utilizados no SSR. É muito importante e foi adotado neste trabalho que a topologia da

rede Zigbee dispense redundâncias de dados, a fim de se obter maior desempenho em meio de intervalo de aquisição de amostras de dados.

Enfatizando a característica modular do SSR na integração de sensores, a possibilidade de independência de cada sensor é bem vinda, assim a metodologia de desenvolvimento do *hardware* se baseia inicialmente em etapas para cada tipo de sensor, ou seja, cada sensor pode ser desenvolvido e integrado separadamente dos outros sensores no sistema.

Em seguida, foi projetada a integração com o microcontrolador, disponibilizando um esquemático elétrico que resultou em um leiaute de circuito impresso e, em seguida, em uma placa protótipo para o elemento móvel do SSR.

Tendo em mãos a placa do elemento móvel, foi necessário confeccionar as estações referência, que contam com suportes do tipo tripé para assegurar uma altura dos sensores acima do solo. Isso foi necessário para se evitar problemas de propagação das ondas eletromagnéticas do ZigBee, pois o solo interfere na propagação do sinal. O *hardware* das estações de referência tem muita similaridade com o do elemento móvel, pois integra os mesmos tipos de sensores, conseqüentemente, somente algumas pequenas alterações no leiaute do circuito foram necessárias, o que facilitou a sua confecção.

Em termos de aplicação, o necessário para o usuário final visualizar as informações adquiridas e mensuradas pelo SSR é um computador com um dispositivo ZigBee, adaptado à porta USB. A conexão fornece os dados a um programa desenvolvido para a geração de gráficos e análises matemáticas.

Todos os *softwares* e itens propostos necessários no desenvolvimento do *hardware* do SSR são descritos a seguir.

- **Isis 7 Professional** é destinado aos usuários na confecção de esquemáticos elétricos, permite uma rápida composição de novos dispositivos eletrônicos, é oferecido pela LabCenter®. [11]
- **Ultiboard 11** é um programa oferecido pela National Instruments® para a confecção de leiautes de circuito impresso (PCB). [12]
- **PCWHD IDE Compiler** da CCS® é um IDE e compilador de códigos na linguagem C para microcontroladores da Microchip®. [13]

- **MPLAB X IDE** oferecido pela Microchip® é destinado em conjunto com gravadores PICKit3® também do fabricante, para a escrita de códigos compilados nos microcontroladores. [14]

Os *softwares* a seguir são relacionados aos sensores e dispositivos do SSR.

- **Visual GPS Application** da VisualGPSTTM compõe gráficos de latitude e longitude de dispositivos GPS que respeitam o padrão de comunicação NMEA. O *software* também permite a gravação em tempo real de dados do dispositivo em arquivos de texto. [15]
- **X-CTU** fornecido pela Digi International® compõe uma interface para a configuração de módulos Zigbee, por meio de comandos seriais ou pela porta USB através de adaptadores. [16]

Ao longo do desenvolvimento do SSR foram necessárias ferramentas matemáticas para constatação das informações e disponibilização de resultados visuais no computador. Assim o *software* MATLAB da Mathworks® [17] foi o ambiente escolhido para manipulação das várias variáveis presentes nos dados fornecidos pelo SSR.

Para facilitar a utilização dos dados de GPS foi criado um programa em Pascal no ambiente de desenvolvimento Delphi ® 2010 da Embarcadero®, fazendo com que o GPS do SSR disponibilize de maneira enxuta as informações dos atletas, durante a prática das modalidades esportivas aos treinadores, que no caso são os testes de validação.

Com o *hardware* do sistema pronto, elemento móvel e estações fixas, o desenvolvimento se focou na interpretação dos dados sensoriais e composição de algoritmos para o tratamento e processamento das informações. Esta etapa sugeriu pequenas alterações no *hardware*, como a alteração do *Baudrate* do módulo de GPS, o que já é previsto no desenvolvimento, portanto, não houve demais complicações na modificação do sistema.

Por fins experimentais foram desconsideradas as dimensões dos dispositivos confeccionados sendo que, uma vez definidos, tem-se a possibilidade de ser feita a miniaturização dos dispositivos através da utilização de tecnologias de montagem SMT e 3D. Isso permite aumentar o conforto dos atletas, dado que os elementos móveis devem ser acoplados de forma a

não prejudicar a prática esportiva. A miniaturização, caso realizada, exigirá novo desenho de leiaute e aquisição de componentes SMD e não foi abordada neste trabalho.

O produto confeccionado, ou seja, o SSR foi testado em ambientes controlados para a sua verificação e demonstração, definindo suas características de limites de instrumentação e o possível enquadramento das modalidades esportivas.

4. Composição do sistema

O GPS é uma opção para o monitoramento de atletas ao ar livre, porém é caro com o aumento da exatidão e não funciona de maneira aceitável em ambientes fechados. O padrão de comunicação ZigBee oferece recursos, como visto na Seção 2.4, que podem ser usados para a determinação da posição de estações base, sendo uma boa opção para sistemas de posicionamento em ambientes fechados, além da possibilidade da troca de dados de uma rede de sensores.

4.1 Dispositivo Móvel

O diagrama apresentado a seguir refere-se ao modelo de *hardware* de uma estação móvel, que é acoplada a um atleta, portanto, pode sofrer alterações caso seja uma estação de referência fixa para a aferição dos valores RSSI ou de uma estação de gerenciamento dos comandos AT dos dispositivos ZigBee, os comandos AT são bem parecidos com os comandos de configuração de *modems* que são enviados através de uma interface serial e que seguem um padrão tabelado [18]. O sistema composto pode ser visualizado na Figura 4.1.

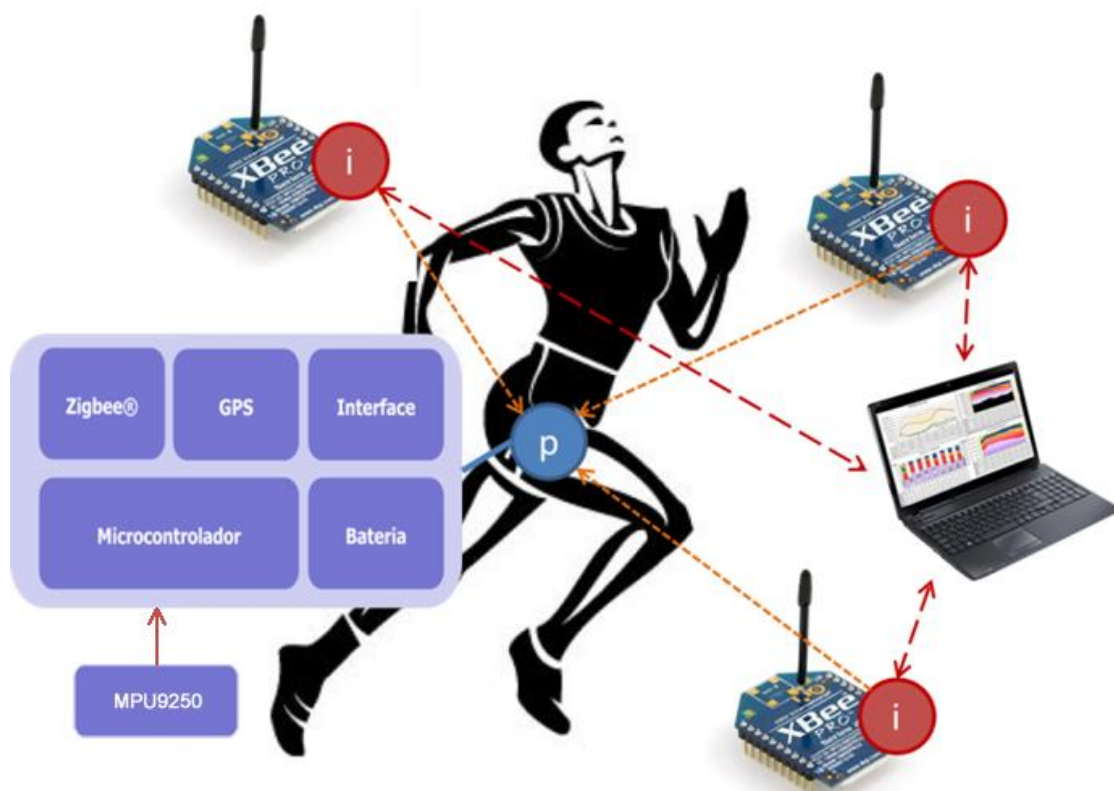


Figura 4.1- Sistema de rastreamento de atletas proposto, visão geral.

Com uma visão do sistema, a Figura 4.1 demonstra o funcionamento do sistema ao seu todo, onde os pontos “i” em vermelho são as estações de referência de trilateração que comunicam com o computador, e o ponto “p” o dispositivo móvel acoplado no atleta. Na Figura 4.1 o dispositivo móvel é expandido em um bloco, que descreve os elementos que o compõe, ou seja, os sensores presentes e o *hardware*.

Baseado na disposição dos sensores e recursos necessários a serem implantados no dispositivo móvel, através dos *softwares* Ultiboard [12] e Isis [11], foi confeccionado um leiaute de placa de circuito impresso, vide Figura 4.2, e então confeccionada uma placa de protótipo do dispositivo móvel que é acoplada ao atleta, vide Figura 4.3.

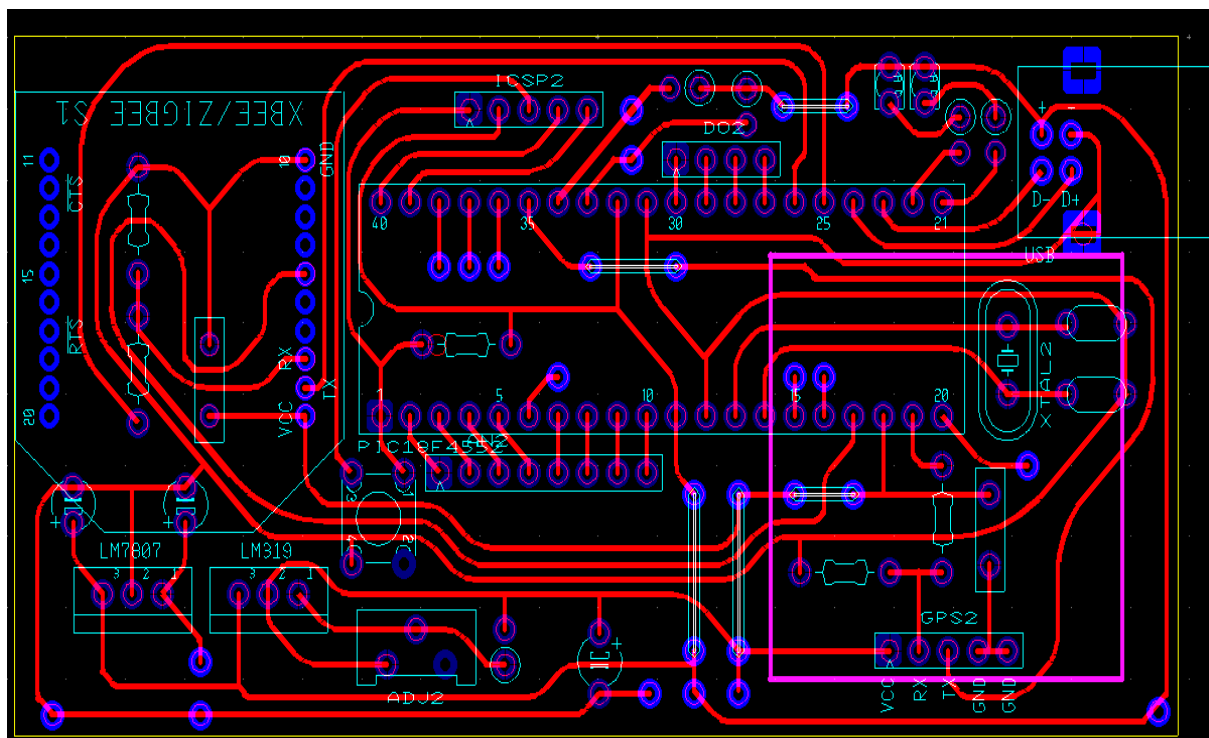


Figura 4.2 - Leiaute de PCB do dispositivo móvel.

O Leiaute foi desenhado manualmente para melhor uso do espaço da placa, numa possível miniaturização será necessário redesenhá-lo para componentes do tipo SMD, porém sempre respeitando o esquemático de ligação dos componentes.

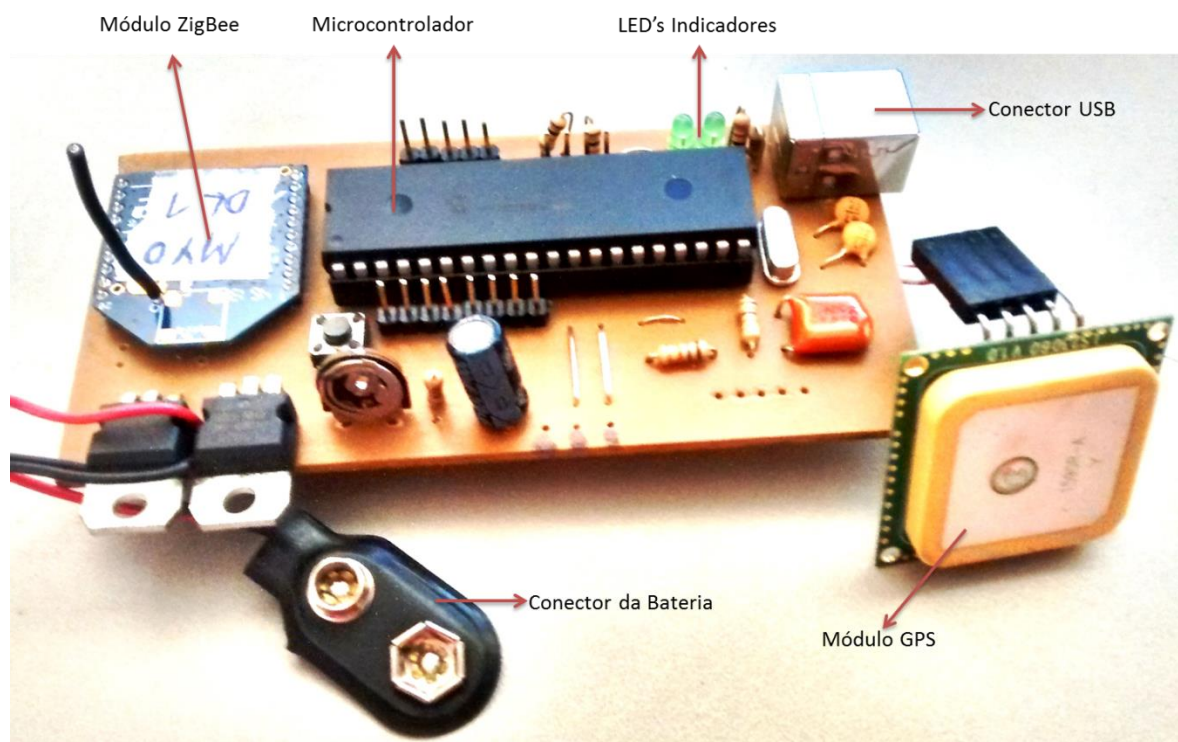


Figura 4.3 - Placa confeccionada do dispositivo móvel.

A placa confeccionada conta com um encapsulamento de espuma e caixa patola para acoplamento no atleta através de um cinto de suporte, ficando localizado nas costas do indivíduo. Porém é necessário ainda um estudo sobre o posicionamento do dispositivo, a fim de evitar o incômodo e não atrapalhe na prática esportiva.

4.2 Estações fixas para Trilateração

As estações que ficam como base referencial para os dispositivos móveis na determinação dos valores de RSSI e, em seguida, a aplicação da trilateração. Conta com suportes feitos de cano PVC com 1,8 m de altura, vide Figura 4.4. O leiaute e as placas dos dispositivos fixos são os mesmos dos dispositivos móveis, porém não são soldados os sensores ou o GPS.

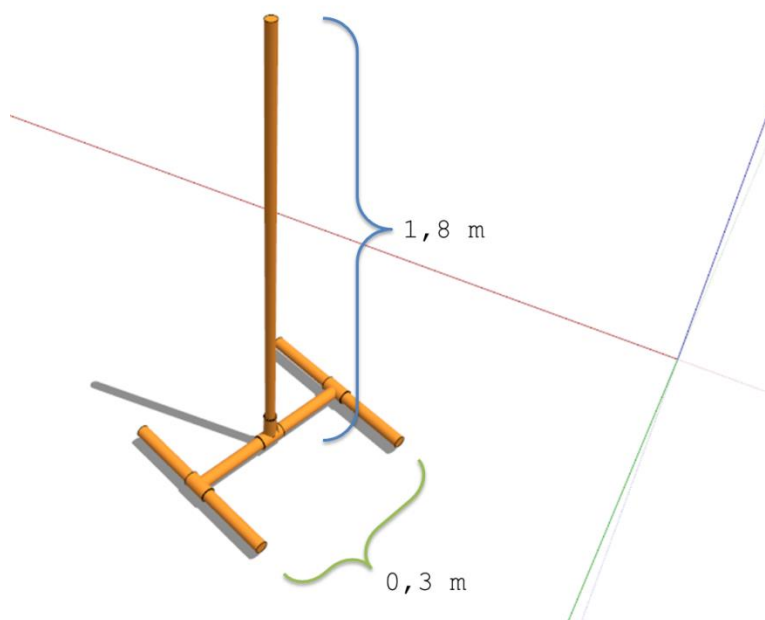


Figura 4.4 - Suporte individual das estações de referência, desenho em 3D.

Nos suportes, as placas são fixadas através de cintas plásticas de fácil remoção, como são fixadas em posições específicas, as estações de referência podem ter a alimentação por fio, mas a aplicação pode sugerir o uso de baterias, o que não é evitado pelo modelo do suporte.

4.3 Propriedades do módulo GPS.

O módulo GPS, LS20031, será integrado nas estações móveis é fabricado pela LOCOSYS Technology Inc., este é representado na Figura 5.1 e possui características de funcionamento descritas pela Tabela 4.1.

Tabela 4.1 - Propriedades do módulo GPS LS20031. [19]

Chip	MediaTek MT3339	
Frequency	L1 1575.42MHz, C/A code	
Channels	Support 66 channels (22 Tracking, 66 Acquisition)	
Update rate	1Hz default, up to 10Hz	
Acquisition Time	Hot start (Open Sky)	< 1s (typical)
	Cold Start (Open Sky)	32s (typical) without AGPS
		<15s (typical) with AGPS (hybrid ephemeris prediction)
Position Accuracy	Autonomous	3m (2D RMS)
	SBAS	2.5m (depends on accuracy of correction data)
Datum	WGS-84 (default)	
Max. Altitude	< 50,000 m	
Max. Velocity	< 515 m/s	
Protocol Support	NMEA 0183 ver 3.01	9600 bps ⁽¹⁾ , 8 data bits, no parity, 1 stop bits (default)
		1Hz: GGA, GLL, GSA, GSV, RMC, VTG

4.4 MPU-9250

O MPU-9250, vide Figura 4.5, é um módulo multi-chip (MCM) consistente de duas matrizes integradas em um único encapsulamento. Uma das matrizes abriga um Acelerômetro de 3-eixos e um Giroscópio de 3-eixos. A outra matriz abriga o chip AK8963, um magnetômetro de 3-eixos da *Asahi Kasei Microdevices Corporation*. Assim, o MPU-9250 é um dispositivo de rastreamento de movimento de 9-eixos que combina um giroscópio de 3-eixos, acelerômetro de 3-eixos, magnetômetro de 3-eixos e um processador digital de movimento (DMP), tudo em um pequeno encapsulamento de medidas 3x3x1mm. Com seu barramento I2C integrado, o MPU-9250 diretamente provê saída completa de 9-eixos. O MPU-9250 também conta com porta I2C extra, para sensores não inerciais extras, como de pressão e temperatura. O chip conta com conversores analógico-para-digital de 16 bits de resolução para todos os eixos de sensoriamento. Para rastreamento de precisão, de ambos os movimentos, rápidos ou lentos, o chip conta com escalas programáveis de cada um dos três sensores integrados, respectivamente, alcance de ± 250 , ± 500 , ± 1000 , e $\pm 2000^\circ/\text{sec}$ (dps) para o giroscópio, $\pm 2g$, $\pm 4g$, $\pm 8g$, e $\pm 16g$ para o acelerômetro, e uma escala de $\pm 4800\mu\text{T}$ para o magnetômetro.

Este chip de última geração foi escolhido por integrar todos os tipos de sensores inerciais, facilitando a interligação com o microcontrolador, consequentemente reduz custos operacionais, igualmente oferecendo baixo custo e menor tamanho físico. [10]

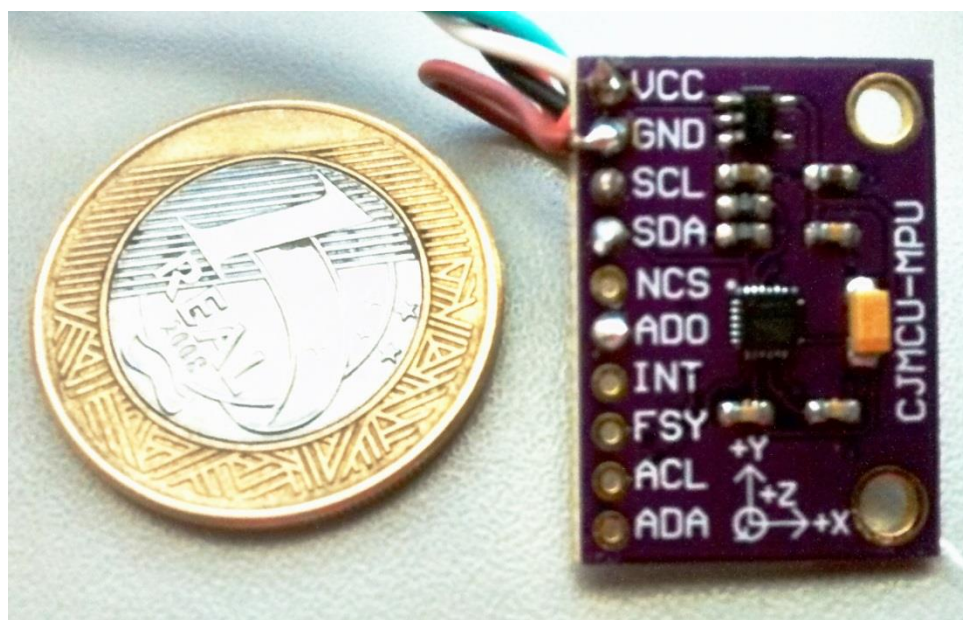


Figura 4.5 - Sensor MPU 9250.

4.5 Propriedades físicas do módulo ZigBee.

Para maior conforto do atleta é necessário que a plataforma tenha o menor tamanho possível, e que também seja muito leve, pois não deve incapacitar qualquer tipo de movimento do atleta durante a execução da modalidade esportiva. Também deve ser resistente a diversos fatores, como, pancadas, líquidos, sacudidas, entre outros. A Figura 4.6 e a Figura 4.7 representam, respectivamente, características mecânicas do módulo ZigBee com visão de cima e visão lateral. Adicionando o módulo GPS pode-se considerar que a placa do dispositivo tem cerca de 100mm x 50mm, de largura e altura, respectivamente. O tamanho da placa depende da tecnologia de confecção utilizada, por exemplo, *Surface Mount Technology* (SMT) e *Trough Hole Technology* (THT), e o projeto físico da disposição dos componentes. Em alguns testes foram utilizadas matrizes de contato para a prototipagem.

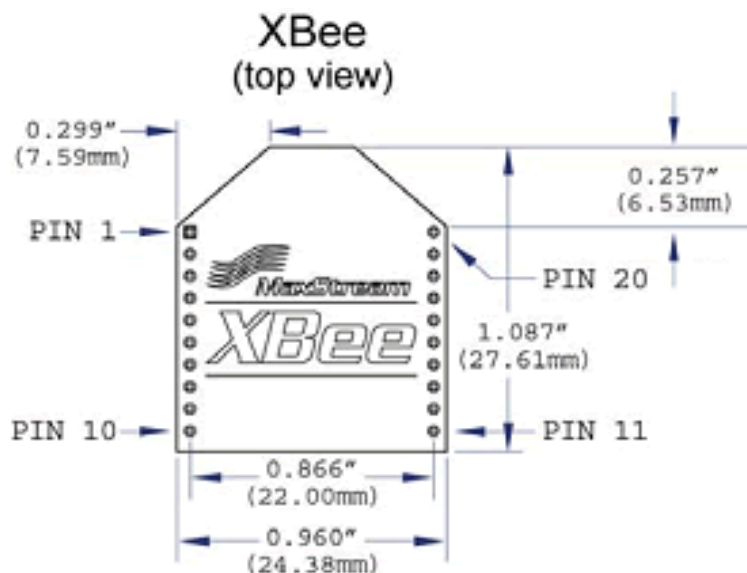


Figura 4.6- Módulo IEEE802.15.4/ZigBee, vista de cima, propriedades mecânicas. [20]

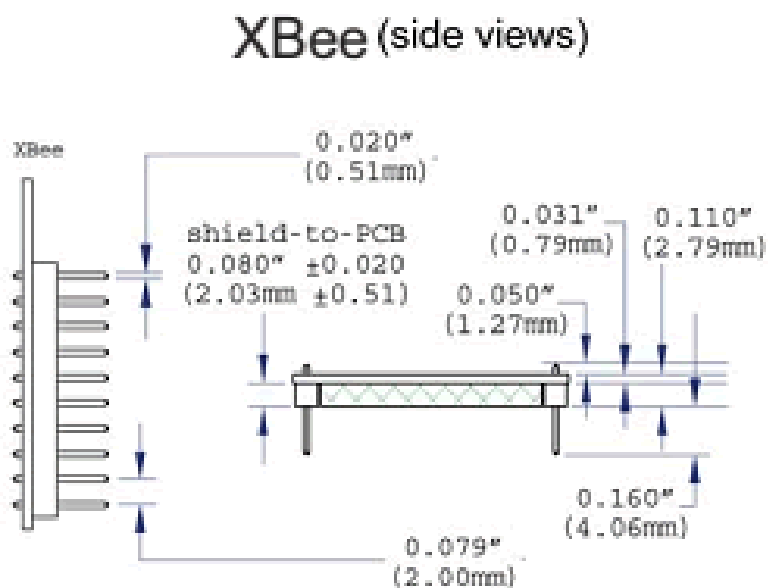


Figura 4.7- Módulo IEEE 802.15.4/ZigBee, vista lateral, propriedades mecânicas. [20]

Nas propriedades mecânicas dos módulos não existem especificações sobre a antena, pois existem quatro tipos de antena para os módulos, a antena de microfita, antena externa com conector RPSMA, a antena do tipo *wire* que é maleável e soldada ao módulo e a antena com

conecto do tipo U.F.L. de padrão japonês. É interessante o fato de uso de conectores, pois se pode futuramente efetuar outros estudos para projetos de antenas adaptáveis ao corpo do atleta.

4.6 Considerações sobre a estrutura física do sistema

A bateria empregada na plataforma deve ser capaz de prover energia suficiente para o funcionamento durante toda a modalidade do esporte, a fim de se evitar maiores custos com manutenção e para não atrapalhar a prática da modalidade esportiva. Deve ser leve e recarregável, uma boa opção, seria a utilização de baterias de Lítio e polímeros, onde atende todos os requisitos, oferecendo mais confiabilidade ao sistema.

Devido ao alto custo das baterias de Lítio, os testes foram executados com baterias recarregáveis comuns.

4.7 Interface de sensores no sistema.

Como a plataforma conta com um microcontrolador PIC para a manipulação dos módulos ZigBee por comandos AT, estes também são utilizados para a aquisição de dados de outros tipos de sensores, com utilidade julgada pela modalidade esportiva. Esses sensores podem possuir interface digital ou analógica e podem facilmente ser implantados, como por exemplo, sensores de temperatura, acelerômetros, *strain-gauge*, magnetômetros, giroscópios, medidores de frequência cardíaca, entre outros. Os sinais desses sensores fazem parte da mesma rede WSN e não interferem no processo de rastreamento. Para que isso seja possível é necessário multiplexar as estações móveis em relação ao canal de comunicação bidirecional com a estação gerenciadora da rede. A multiplexação deve ser do tipo de divisão temporal (TDM), pois o ZigBee já tem frequência estabelecida. A execução da técnica de multiplexação também é tarefa do microcontrolador.

Para o caso da rede WSN, o tempo da divisão da multiplexação não é fixo e é controlado pela estação de gerenciamento, funciona pela requisição de valores dos sensores da rede interligados nos nós móveis, é um valor dinâmico que dependerá do tempo resposta e quantidade de dados disponíveis para a leitura da estação de gerenciamento, características do DTDM.

4.8 Topologia de rede do sistema

Para assegurar a disponibilidade dos dados sensoriais e o mínimo de fluxo na rede, a topologia de rede do sistema ZigBee deve considerar as limitações do ZigBee em termos de taxa de dados. Desse modo, para evitar problemas em altas de taxas de aquisição de dados sensoriais, a topologia utilizada é mostrada na Figura 4.8. Observa-se que possui coordenação pelo computador (PC) que se comunica indiretamente com o dispositivo móvel (M) por intermediação das estações de referência (R).

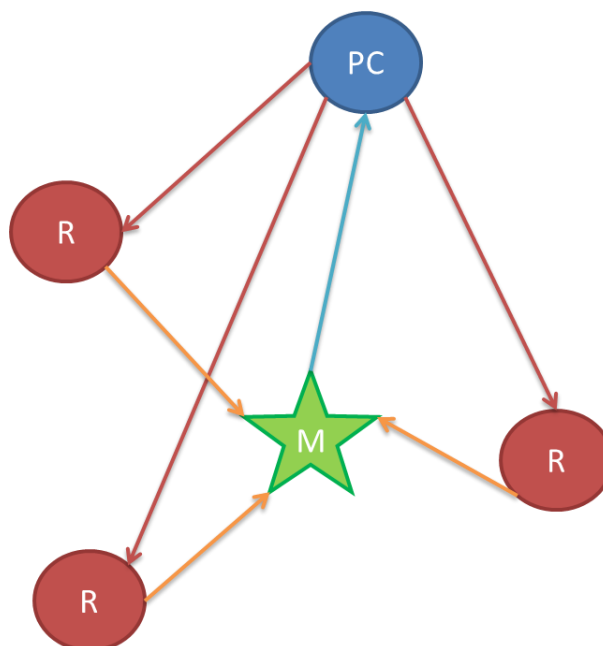


Figura 4.8 - Topologia de rede do Sistema.

Dado o sentido unidirecional da informação entre os dispositivos, a topologia evita a bidirecionalidade da informação, assim aproveitando o máximo de taxa possível oferecida pelo ZigBee, ganhando em desempenho na taxa de aquisição de dados sensoriais dos dispositivos móveis. No caso da Figura 4.8 somente é exibido um elemento móvel da rede, mas isso não descarta a possibilidade de utilização de mais dispositivos, como no caso de monitoramento de vários atletas no mesmo instante.

5. Testes e Resultados

Para averiguação do funcionamento das tecnologias ZigBee e GPS foram elaborados uma série de experimentos que fazem testes com uma versão protótipo do sistema, essa versão foi montada em matrizes de contato, logo, foi confeccionada uma PCB.

5.1 Experimento de análise do padrão de erro em posicionamento absoluto por código (C/A)

Para analisar o receptor de GPS por código (C/A) foi realizado um experimento simples de aquisição de coordenadas para verificação da existência de um padrão de erro neste tipo de GPS.

5.1.1 Metodologia

- Foram adquiridos dois módulos GPS por código (C/A) do fabricante LOCOSYS® de modelo LS20031, vide Figura 5.1.
- Ambos foram conectados num computador portátil através de duas placas de desenvolvimento XBEE™ da Digi®, que possuem conversores RS-232/USB.
- Utilizando o *software* “VisualGPS Application” da Visual GPSTM, foi feita a gravação dos dados dos dois módulos GPS, este *software* realiza médias e plotagens referentes às coordenadas obtidas, vide Figura 5.2.
- A aquisição dos dados foi feita simultaneamente e com ambos dispositivos estáticos.
- Os gráficos dos traços da trajetória de todos os GPS foram comparados.

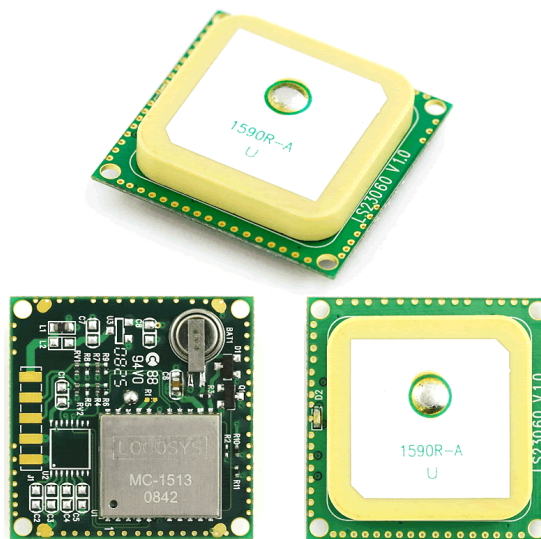


Figura 5.1 – Módulo GPS LOCOSYS® LS20031. [19]

Os módulos GPS transmitem informações no formato NMEA, que são textos com dados de coordenadas, satélites utilizados, intensidade de sinal com os satélites, velocidade referencial, entre outros. Estes dados são interpretados pelo *software*, vide interface gráfica mostrada na Figura 5.2, e convertidos em informação visual.

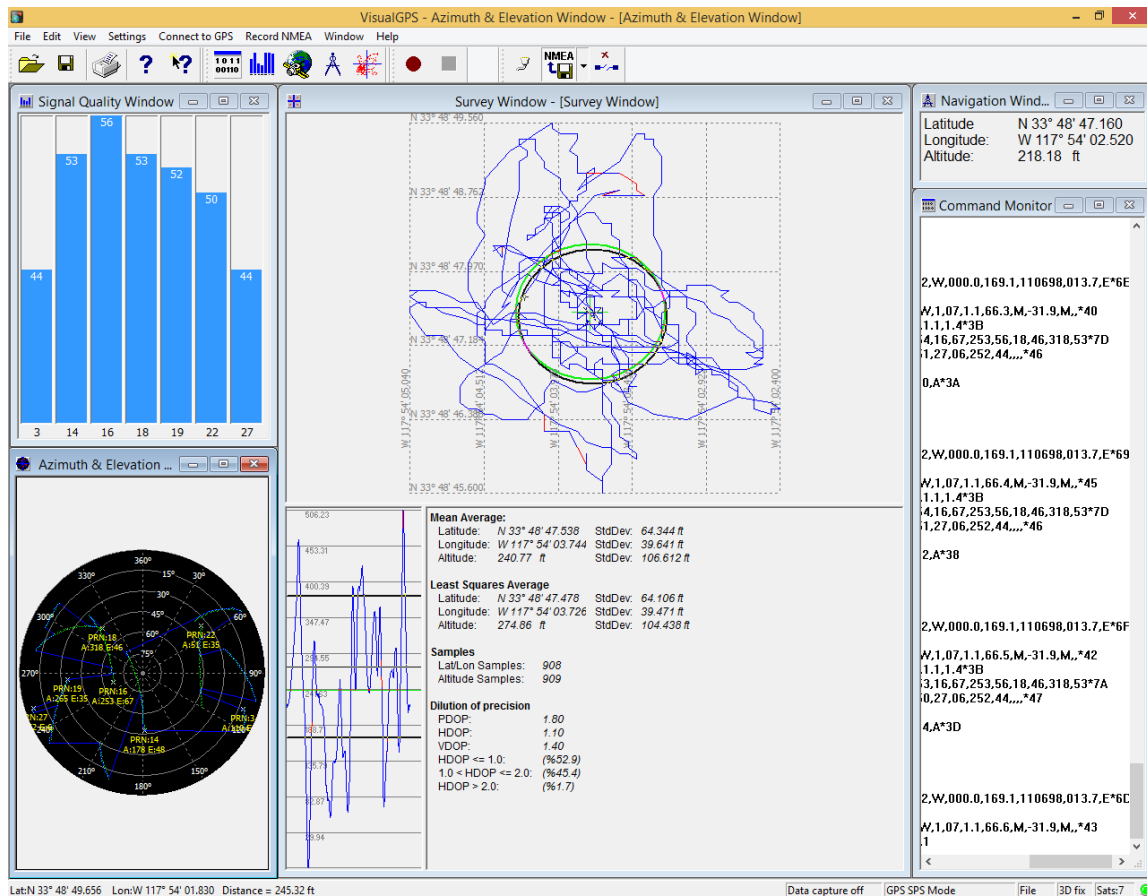


Figura 5.2 – Software “Visual GPS Application”, da Visual GPSTM. [15]

5.1.2 Resultados

A Figura 5.3 exibe um gráfico das coordenadas adquiridas pelo GPS número 1. Percebe-se que mesmo estando estático, existe uma variação muito grande na posição, sendo os eixos de latitude e longitude a grade da figura, a linha azul a trajetória desenhada pelo *software* de acordo com as coordenadas adquiridas, a elipse verde com uma cruz no meio é a média de todas as amostras e a elipse preta com uma cruz, a média dos quadrados mínimos.

Percebe-se que neste GPS, a elipse verde está estendida verticalmente. Conclui-se que a latitude sofreu maiores variações que a longitude.

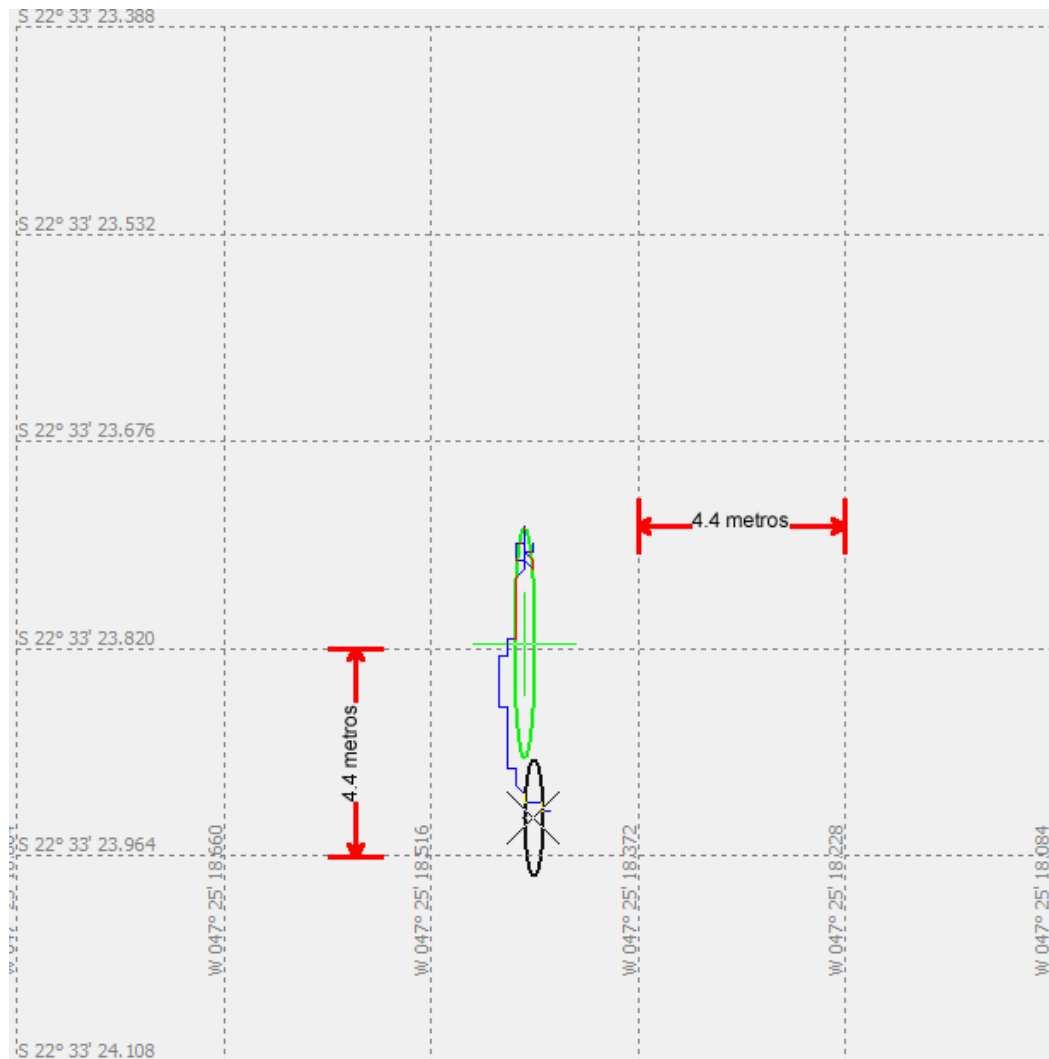


Figura 5.3 - Gráfico resposta do GPS 1.

A Figura 5.4 exibe um gráfico das coordenadas adquiridas pelo GPS número 2, onde também foram encontradas variações mesmo estando estático. A mesma condição ocorre no GPS 1, que também apresenta uma variação significativa na posição. Percebe-se que nesse GPS, a elipse verde está estendida horizontalmente e conclui-se que a longitude sofreu maiores variações que a latitude.

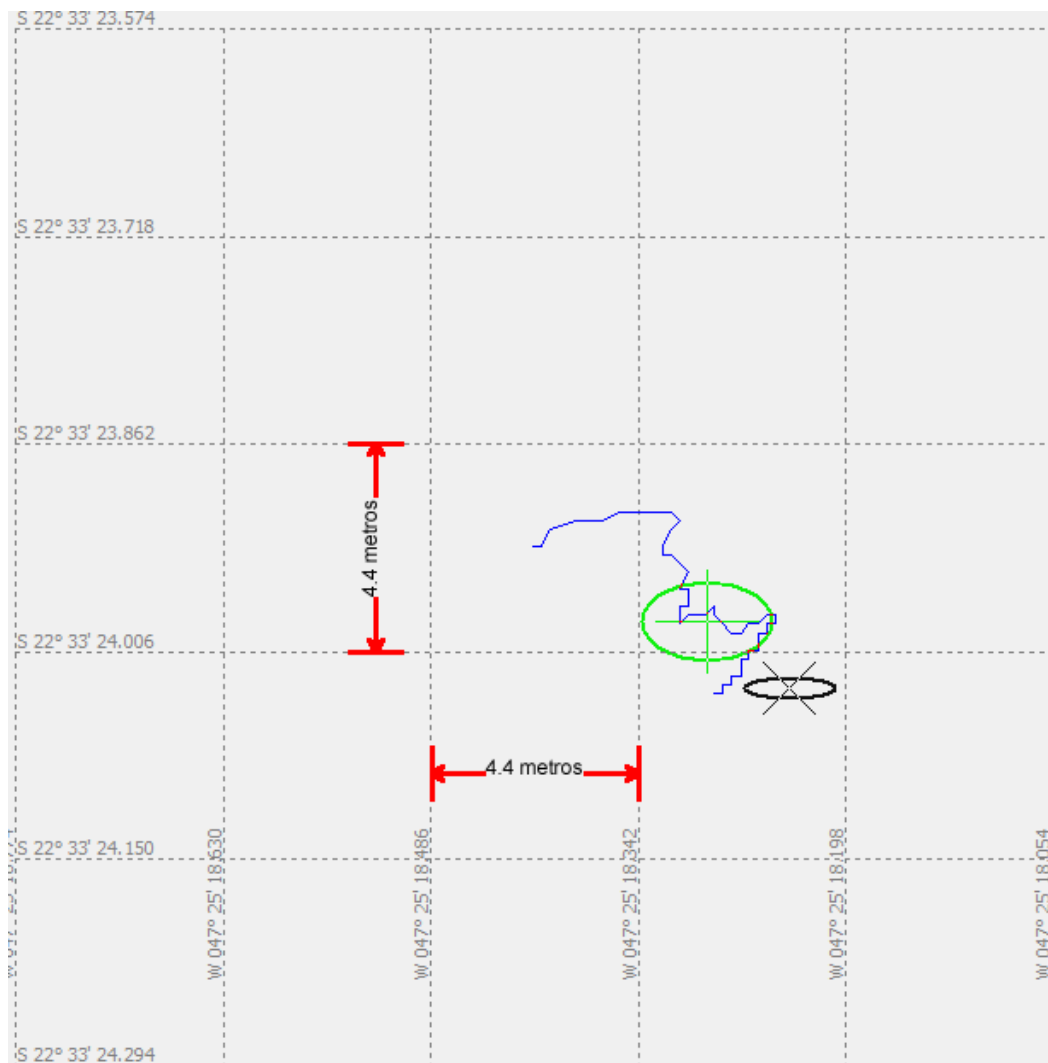


Figura 5.4 - Gráfico resposta do GPS 2.

Tendo-se os gráficos do GPS 1 e 2, pode-se afirmar que mesmo sob as mesmas condições de funcionamento, como, modelo, posição, tempo de aquisição e horário da aquisição, o GPS por código (C/A) não apresenta nenhum tipo de erro padrão que seja utilizável como referência entre os dispositivos para correções de posição. Uma vez que o GPS 1 apresentou resultados com maiores variações na latitude e o GPS 2 na longitude, consequentemente, se subtraída a trajetória de um com o outro, não haverá um ponto estático como resposta e sim uma outra trajetória.

Embora o GPS de código (C/A) não apresente padrões de variações posicionais, ele ainda apresenta a sua característica de baixo custo e, dependendo da aplicação, é aceitável o erro de exatidão que esse tipo de GPS possui.

5.2 Experimento comparativo de curvas de RSSI aferidas e teórica.

Para visualização do fenômeno de atenuação da energia de propagação de um sinal pela distância de forma exponencial, foi feito um experimento em ambiente controlado descrito a seguir.

5.2.1 Metodologia

No mês de novembro de 2013, em uma quadra poliesportiva coberta, do Clube de Associados AABB do Banco do Brasil, situado na cidade Limeira, estado de São Paulo, foi utilizada uma linha lateral da quadra como referência de distância entre um módulo ZigBee estático, de referência, fixado numa extremidade da linha e um módulo ZigBee ligado a um computador. A linha foi seccionada com fita marcadora de forma que entre o zero e um metro de comprimento existisse marcações de 10 em 10 centímetros, entre um metro e dez metros, marcações de 20 em 20 centímetros, entre dez metros de comprimento e quinze metros, marcações de 50 em 50 centímetros e a partir de quinze metros até toda a extensão de trinta metros da linha, marcações de 1 em 1 metros.

No ponto zero da linha da quadra, foi fixado um módulo ZigBee, interligado a um microcontrolador PIC18F4550 da Microchip®, como pode ser visto na Figura 5.5, que é uma foto tirada para melhor entendimento. Este módulo ZigBee fixo ao receber comandos da estação móvel que foi posicionada em cada marcação ao longo da linha, como visto na Figura 5.6, responde com a mesma informação recebida a estação móvel interligada a um computador, ao que neste, através de um *software*, vide Figura 5.7, desenvolvido no programa IDE Delphi 2010 da Embarcadero®, envia os comandos e os recebe com o ZigBee ligado na interface USB. O computador foi utilizado para a requisição de quatro valores de RSSI do módulo ZigBee acoplado na porta USB, em cada marcação da quadra foi feita uma média aritmética dos valores, através do *software* Excel da Microsoft® e os dados da planilha do Excel foram importados pelo *software* MATLAB para a confecção do gráfico comparativo das curvas dos valores de RSSI aferidos e a teórica.



Figura 5.5 - Foto da metodologia utilizada, vide estação fixa e marcações na linha.

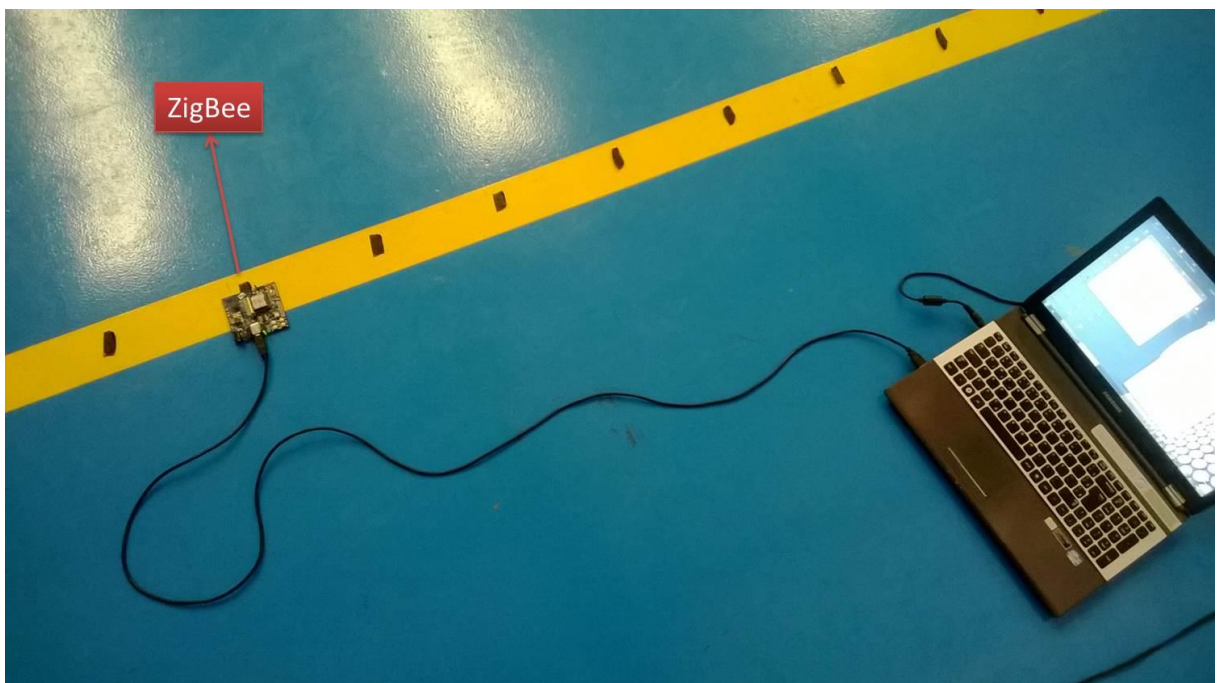


Figura 5.6 - Computador adquirindo valores de RSSI do módulo ZigBee em uma marcação.

No gráfico gerado pelo MATLAB, existem duas curvas de valores RSSI aferidos e a curva teórica, a segunda curva aferida é resultado da mudança de altura dos módulos ZigBee, sendo que ambos ficam em cima de uma mesa de cerca de 1 metro de altura, esta mudança tem como objetivo, a melhora na propagação do sinal e consequentemente um resultado mais próximo ao teórico.

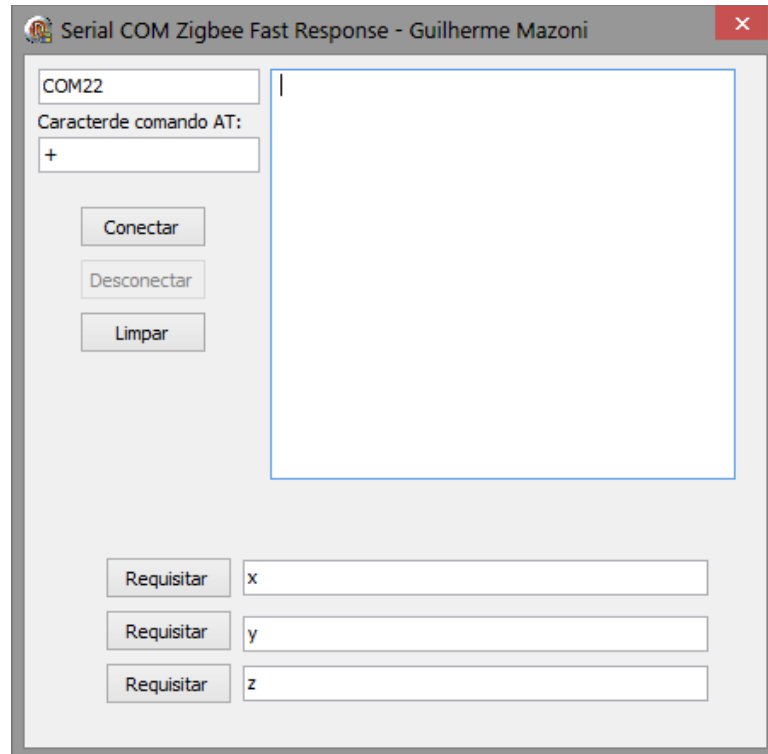


Figura 5.7 - Imagem do software desenvolvido com funcionamento no computador para aquisição dos valores de RSSI da estação móvel.

5.2.2 Resultados

A Figura 5.8 é o gráfico gerado pelo MATLAB com as curvas de RSSI aferidas com as estações em alturas diferentes e a curva teórica. Pode-se observar que a curva obtida através da coleta de dados do ZigBee fixado no chão, somente segue a tendência da curva teórica até cerca de 3 metros de distância. A partir dessa distância os valores são praticamente aleatórios e não podem ser utilizados, ou seja, caso sejam aplicados na Fórmula de Friis para a determinação da

distância. A curva de valores RSSI aferidos que mais tendeu a curva teórica foi na qual o ZigBee ficou a cerca de 1 metro de distância do chão. Nessa curva pode-se visualizar que até um pouco mais de 10 metros, ela seguiu a tendência da teórica, sendo assim, aplicável na Fórmula de Friis até certa distância.

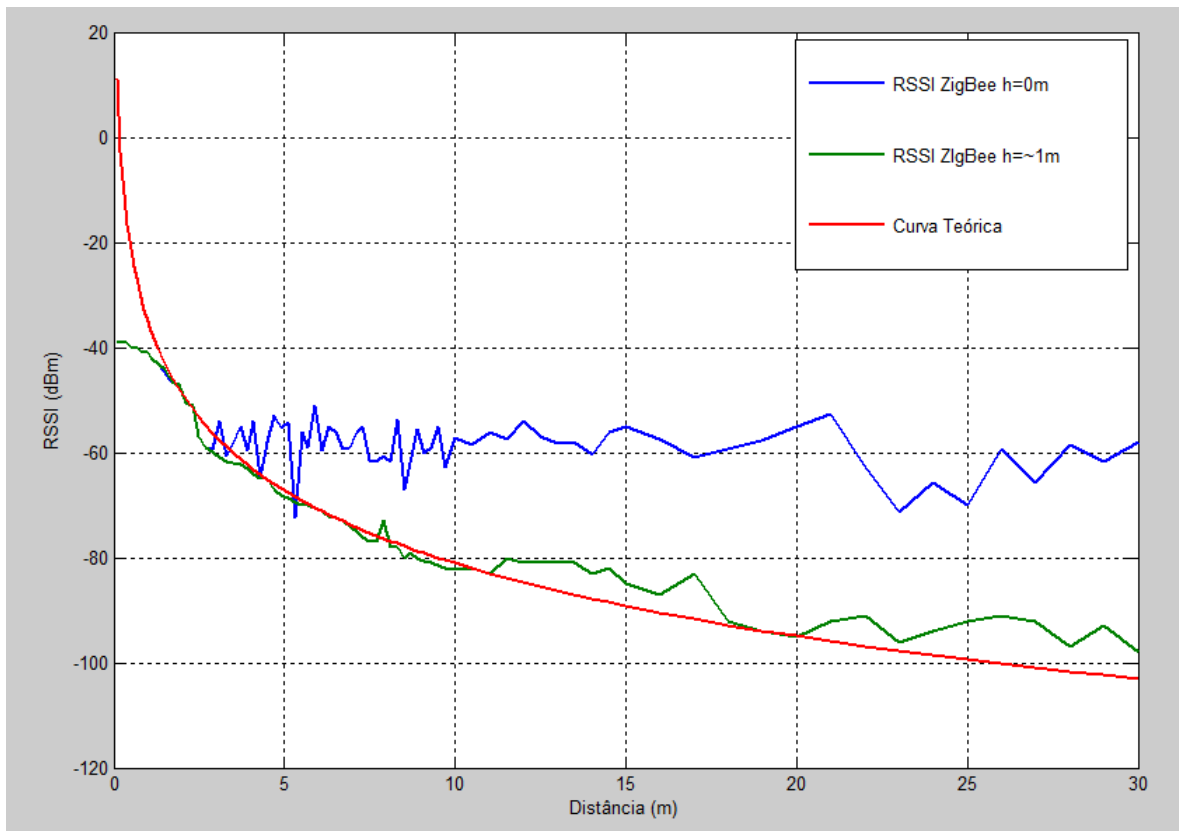


Figura 5.8 - Gráfico comparativo entre as curvas de valores RSSI e a teórica.

É válido lembrar que para a adequação da curva teórica, o valor de k presente na equação obtida através da Fórmula de Friis (**Erro! Fonte de referência não encontrada.**) foi adaptado, pois k é uma constante de calibração na equação, sendo $k = -35$. Na Figura 5.8 em azul, a curva de valores RSSI aferidos com os módulos ZigBee no chão, em verde a curva aferida com os módulos em cima de uma mesa, vide Figura 5.9 e em vermelho a curva teórica.



Figura 5.9 - Módulo ZigBee em cima de uma mesa para melhores condições de propagação dos sinais.

5.3 Experiência com o método de Trilateração com o MATLAB

Dada às equações na Seção 2.6, foi feito um algoritmo no MATLAB para a aplicação de uma versão adaptada da **Erro! Fonte de referência não encontrada.**, onde se resolve as atrizes pela regra de Cramer [4], o algoritmo pode ser visto a seguir dentro do Quadro 5.1.

No experimento foi usada uma metodologia similar ao experimento descrito na Seção 5.2, na mesma quadra, onde foram fixados no chão, três módulos ZigBee, um em cada vértice das linhas extremas da quadra e ao todo foram testadas seis posições aleatórias na quadra, onde a estação móvel se encontrou no chão, vide Figura 5.10. Em cada posição foram obtidos os valores de RSSI, com referência as três estações fixas, de forma que, em cada uma, vinte deles foram aferidos. Os valores de RSSI de cada estação e posição sofreram um tratamento estatístico de

média aritmética, também em cada posição foi medida a distância real com cada estação fixa através de uma trena à Laser.

Dada a metodologia, tem-se a disposição a informação real da distância de cada posição com cada estação fixa, além das distâncias estimadas pela Fórmula de Friis adaptada pela Equação 5.1 disposta a seguir e também as coordenadas 2D de cada estação fixa. Sendo assim o código em MATLAB da Figura 5.10 aplicável.

Equação 5.1

$$d = e^{-\frac{1}{20} \cdot R - \frac{1}{20} \cdot k}$$

Na Equação 5.1, d é distância estimada, R o valor de RSSI em dBm e k a constante de calibração.

```
function [ x, y ] = two_tri(x1, x2, x3, y1, y2, y3, d1, d2,
d3)

x_n11 = (d1^2 - d2^2) - (x1^2 - x2^2) - (y1^2 - y2^2);
x_n21 = (d1^2 - d3^2) - (x1^2 - x3^2) - (y1^2 - y3^2);
x_n12 = 2*(y2-y1);
x_n22 = 2*(y3-y1);

d11 = 2*(x2-x1);
d21 = 2*(x3-x1);
d12 = 2*(y2-y1);
d22 = 2*(y3-y1);

x_n = [x_n11, x_n12; x_n21, x_n22];
d = [d11, d12; d21, d22];

x = x_n/d;
x = det(x);

y_n11 = d11;
y_n21 = d21;
y_n12 = x_n11;
y_n22 = x_n21;

y_n = [y_n11, y_n12; y_n21, y_n22];
```



```

y = y_n/d;
y = det(y);
end

```

Quadro 5.1 - Código em MATLAB para solução de coordenadas em 2D por trilateração. [4]

O código no MATLAB foi validado através da posição real de número 1, onde se encontra no quarto vértice das linhas da quadra, não ocupado por estações referência, sendo o ponto de origem (0,0) do plano XY, vide “P1 Real” na Figura 5.11.



Figura 5.10 - Foto da quadra com marcações de três posições medidas, sendo a mais evidente a posição 6, perto da linha de arremesso de basquetebol a posição 4 e a posição 2 no vértice das linhas do garrafão de basquetebol.

Dados os parâmetros do código MATLAB, as coordenadas XY das estações fixas e as distâncias, reais e estimadas, uma para cada estação referência com relação às posições aleatórias, foi gerado o gráfico no MATLAB, representado pela Figura 5.11. É possível então visualizar a relação de todas as seis posições aleatórias reais e estimadas, além da disposição das estações fixas e um desenho ilustrativo da quadra.

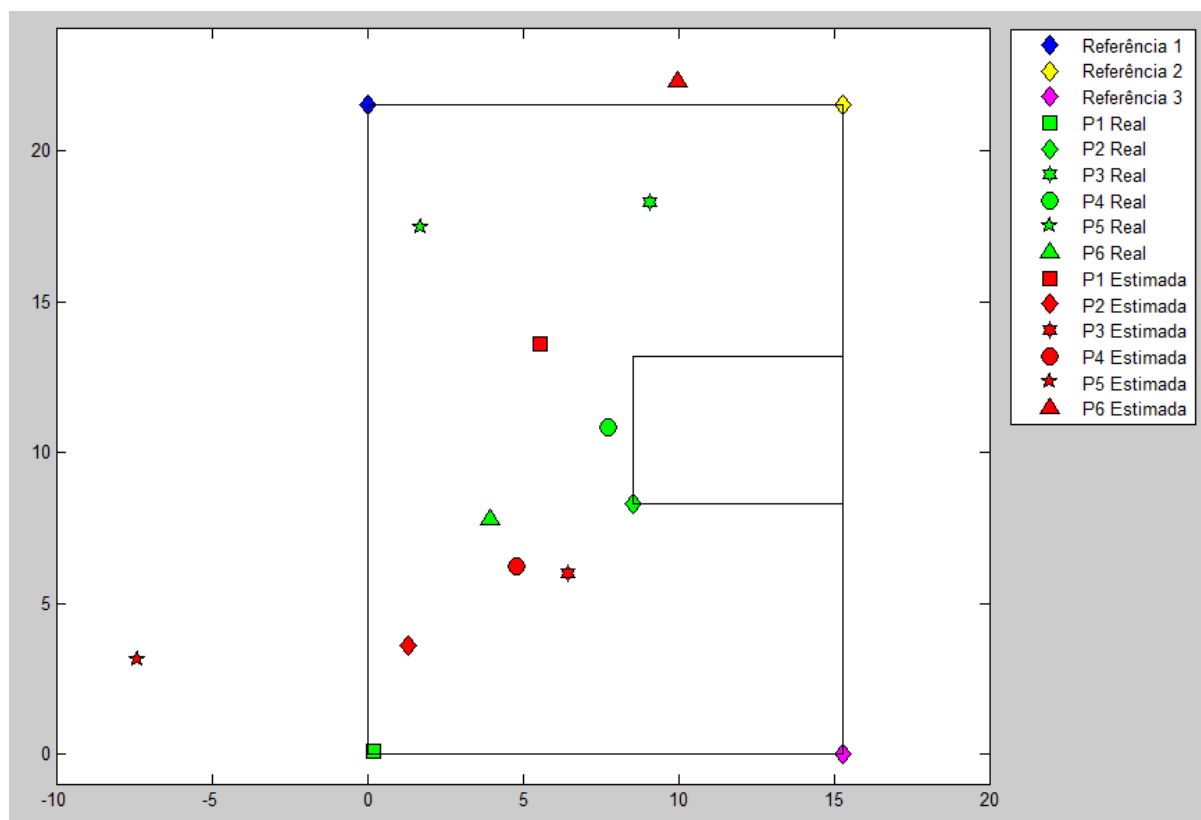


Figura 5.11 - Gráfico com disposição das estações na quadra, compare posições 2,4 e 6 reais com as da Figura 5.10.

No gráfico da Figura 5.11, todas as posições estimadas foram totalmente irregulares, dado que a metodologia, neste experimento constituiu o posicionamento de todas as estações no chão, uma vez que o experimento da Seção 5.2 mostrou que ao posicionar as estações no chão, a estimativa pela Fórmula de Friis não é aceitável, vide Figura 5.8. Assim esse experimento da trilateração foi comprovado somente pelas distâncias reais, uma vez que para a comprovação pelas distâncias estimadas, a metodologia deve ser alterada, ou seja, o parâmetro de altura das estações de referência e da móvel deve ser modificado para um nível aceitável, aonde seja possível oferecer melhores condições de propagação dos sinais da rede ZigBee.

Devido à presença de dados irregulares do teste da trilateração com o Zigbee, como visto na Figura 5.11, foi realizado um novo teste seguindo melhores condições de propagação para o ZigBee como na Seção 5.2. Assim foi elevada a altura dos módulos ZigBee por meio de suportes confeccionados com canos de PVC, que possuem altura de 1,8 m. No novo teste foi utilizada a

mesma quadra, porém com um campo de teste menor e com novas posições aleatórias escolhidas, resultando então na Figura 5.12 a seguir.

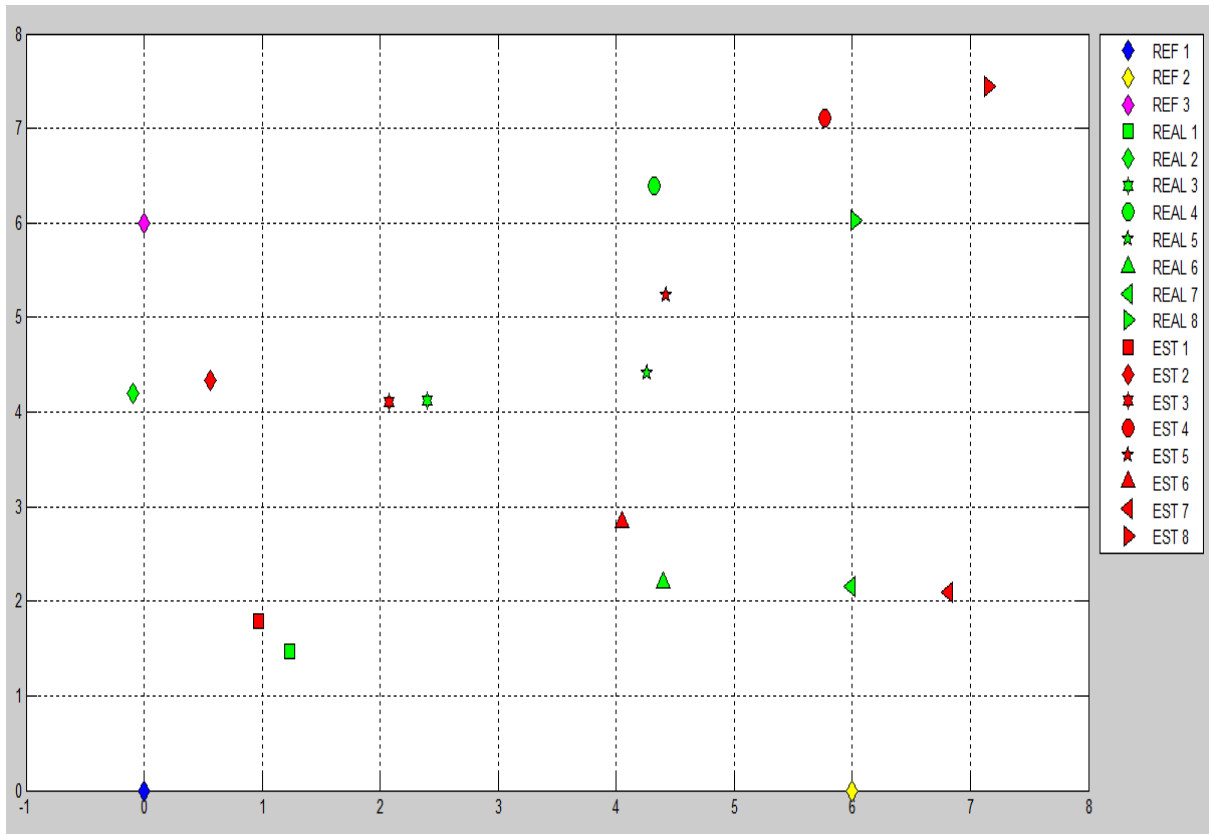


Figura 5.12 - Gráfico com disposição das estações na quadra e valores estimados do novo teste.

A partir da Figura 5.12, todas as posições estimadas foram próximas aos seus valores reais, dado que a metodologia, neste experimento constituiu o posicionamento de todas as estações acima do chão. Assim, pode-se observar que as posições estimadas mais próximas da estação de referência 1 (REF1) oferecem maior exatidão comparando com as mais distantes, pois estão mais próximas da potência do sinal, justificando a curva da Figura 5.8 , onde tem-se maior diferença de valores de potência pela proximidade da antena transmissora. Essa diferença impacta na sensibilidade dos módulos ZigBee na determinação dos valores de RSSI, lembrando que o valor de RSSI possui uma resolução limitada, como visto na Seção 2.4.

Pelo novo teste, pode-se afirmar que é possível obter exatidão na próxima de metros, para um sistema de rastreamento *indoor* com o ZigBee. Essa exatidão pode ser melhorada quando se utiliza distâncias menores entre as estações móveis e a base.

5.4 Experiência de coleta de dados GPS do sistema

5.4.1 Metodologia

No mês de dezembro de 2014, em uma quadra poliesportiva, da Faculdade de Ciências Aplicadas - UNICAMP, situada na cidade Limeira, estado de São Paulo, foram realizados testes com o GPS do sistema. Numa parceria com professores da UNICAMP, especialmente o Prof. Dr. Milton Shoiti Misuta da FCA e o Prof. Dr. Mauro Menzori da FT, as informações resultantes dos testes puderam ser comparadas com as informações de GPS de fase de altíssima precisão e exatidão. Devido à característica de precisão do GPS de fase, como grau comparativo, ele foi considerado como fonte de valores ideais de posicionamento, dado que o erro de exatidão em cm é muito pequeno. Assim, são feitas comparações com o GPS de fase, fornecido pelo Prof. Mauro, e o GPS de código do sistema confeccionado.

A Figura 5.13 representa o esboço da quadra de testes com pontos determinados de “A” a “L” e um ponto de origem “O”, este esboço é tomado como modelo para quatro tipos de percurso. Com o GPS de Fase e o sistema confeccionado, um atleta percorreu os percursos com quatro velocidades diferentes:

1. Andando (0 - 6 Km/h)
2. Trotando (10 Km/h)
3. Correndo (15 Km/h)
4. Sprint (18 Km/h)

Têm-se então, dados de dois tipos de GPS para cada percurso com velocidades diferentes, os percursos. Lembrando que os valores retornados pelo GPS de fase foram considerados ideais.

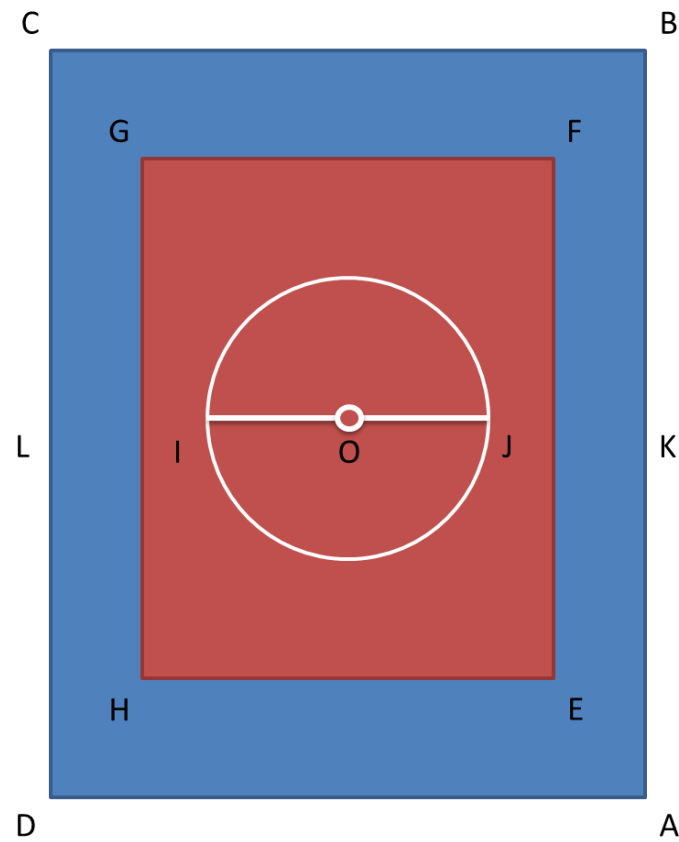


Figura 5.13 - Desenho da quadra de testes.

Os percursos executados foram:

1. $A \rightarrow D \rightarrow C \rightarrow D \rightarrow A$ (Formato em L)
2. $E \rightarrow F \rightarrow G \rightarrow H$ (Retângulo Interno)
3. Círculo de I à J
4. “Zig-Zag” entre os pontos A, E, meridiano de DA, H, D e novamente A.

Devido o surgimento de problemas com a bateria e a transmissão remota dos valores do GPS de código, os resultados incluem somente o segundo e o terceiro percurso, com as velocidades andando e correndo, resultando nos testes descritos através da Tabela 5.1 a seguir.

Tabela 5.1 - Percursos e velocidades com gráficos comparativos.

Percurso: E → F → G → H (2)		Percurso: Círculo de I à J (3)	
Nome	Número	Nome	Número
Andando	6	Andando	8
Andando	7	Andando	9
Correndo	5	Correndo	7
Correndo	6	Correndo	8

O módulo de GPS LS20031 [19] fornece os dados no protocolo NMEA. E, para a análise em plano é necessário extrair da sentença NMEA, a latitude e a longitude e convertê-las para coordenadas planas. Para isso foi usada uma ferramenta WEB [21], vide Figura 5.14, que provê os meios para a transformação entre sistemas de coordenadas. Ao extrair a latitude e longitude do NMEA do GPS de código, têm-se as coordenadas em formato DMM, ou seja, Graus (*Degrees*), Minutos e fração de Minutos. A ferramenta WEB aceita este tipo de entrada e faz diretamente a transformação para o plano UTM (Universal Transversa de Mercator).

Geographic coordinates (Latitude, Longitude)				UTM Coordinates
Hemisphere	DMS	DMM	DDD	
Latitude: <input type="radio"/> N <input type="radio"/> S Longitude: <input type="radio"/> W <input type="radio"/> E	ddd°mm'ss.ss" <input type="text"/> ° <input type="text"/> ' <input type="text"/> "	ddd°mm.mmm' <input type="text"/> ° <input type="text"/> '	ddd.ddddd° <input type="text"/> °	Northing: <input type="text"/> Easting: <input type="text"/> Zone/Sector: <input type="text"/>
*Datum:	WGS84/NAD83 ▼	WGS84/NAD83 ▼	WGS84/NAD83 ▼	WGS84/NAD83 ▼
Magnitude of total shift (WGS84 vs. NAD27): N/A				
- Instructions - 1. Enter the GPS coordinate and the desired datum in one of the columns above 2. Select the hemisphere, if you want to convert Lat/Long values into UTM 3. In case of datum transformation, select the desired datum in the target column(s) 4. Press the convert button				
<div>Convert Reset</div>				
<small>*Note: The datum transformation is an approximation and works for the continental US (CONUS), Alaska and Canada <u>only</u>, because it uses the Molodensky equation and fixed shift parameters for those areas.</small>				

Figura 5.14 - Site conversor de coordenadas. [21]

Mesmo que a ferramenta WEB disponibilize uma maneira de transformação de coordenadas, demandaria demasiado tempo para utilizá-la na conversão em massa de dados num

único percurso, também é limitada na precisão de eixo UTM, dada a transformação *datum* da ferramenta, a entrada dos dados possui somente duas casas decimais por eixo o que também limita a transformação. Por serem dois percursos com duas velocidades de execução cada e repetidos uma vez, no total são oito experimentos que forneceram dados de coordenadas. Foi preciso criar uma ferramenta, desenvolvida no Delphi [22], vide Figura 5.15, que converte em massa as coordenadas de latitude e longitude de um arquivo NMEA.

NMEA to DMS/DMM & DDD - Guilherme Mazoni - 2014

Entrada (NMEA):

```
$GPGGA,140734.688,2233.0608,S,04725.7976,W,1,7,1.01,596.9,M,-4.0,M,,*48
$GPGGA,140734.862,2233.0654,S,04725.8005,W,1,7,1.01,596.3,M,-4.0,M,,*43
$GPGGA,140735.044,2233.0683,S,04725.8024,W,1,7,1.01,595.8,M,-4.0,M,,*4F
$GPGGA,140735.232,2233.0706,S,04725.8036,W,1,7,1.01,596.0,M,-4.0,M,,*48
$GPGGA,140735.422,2233.0723,S,04725.8045,W,1,7,1.01,596.1,M,-4.0,M,,*4D
$GPGGA,140735.615,2233.0736,S,04725.8052,W,1,7,1.01,596.2,M,-4.0,M,,*4A
$GPGGA,140735.809,2233.0746,S,04725.8057,W,1,7,1.01,596.3,M,-4.0,M,,*4A
$GPGGA,140736.005,2233.0752,S,04725.8062,W,1,7,1.01,595.4,M,-4.0,M,,*4A
$GPGGA,140736.199,2233.0759,S,04725.8065,W,1,8,0.99,595.6,M,-4.0,M,,*4F
$GPGGA,140736.387,2233.0766,S,04725.8076,W,1,8,0.99,599.8,M,-4.0,M,,*4E
$GPGGA,140736.539,2233.0863,S,04725.8130,W,1,8,0.99,597.4,M,-4.0,M,,*46
$GPGGA,140736.746,2233.0855,S,04725.8126,W,1,8,0.99,596.9,M,-4.0,M,,*42
$GPGGA,140736.946,2233.0854,S,04725.8125,W,1,8,0.99,595.5,M,-4.0,M,,*41
$GPGGA,140737.147,2233.0853,S,04725.8124,W,1,8,0.99,594.4,M,-4.0,M,,*4F
$GPGGA,140737.347,2233.0854,S,04725.8123,W,1,8,0.99,593.8,M,-4.0,M,,*46
$GPGGA,140737.548,2233.0855,S,04725.8121,W,1,8,0.99,593.6,M,-4.0,M,,*42
$GPGGA,140737.747,2233.0855,S,04725.8120,W,1,8,1.09,593.4,M,-4.0,M,,*44
```

DMS: ddd°mm'ss,sssss"

Latitude: 22 33 03,92400 S

Longitude: 47 25 48,03600 O

Horário: 14:07:34,862

DMM: ddd°mm,mmmm'

Latitude: 22 33,0654 S

Longitude: 047 25,8005 O

DDD: ddd,dddd°

Latitude: -22,551090000

Longitude: -47,43001

Operação do TXT

Exportar DMS Exportar DMM Exportar DDD

Figura 5.15 - Programa desenvolvido no Delphi para conversão de coordenadas GPS.

O Programa converte o arquivo NMEA para os formatos DMM, Graus Minutos e Segundos (DMS) e Graus e Décimos de Graus (DDD), o DMM é obtido somente pela extração do arquivo NMEA, DDD e DMS são calculados a partir do DMM. Com o formato DDD de coordenadas é

possível aplicar um código em MATLAB [23], a função deg2utm, vide trecho do código no Quadro 5.2.

Exemplo de uso da função:

Dados de entrada:

Coordenadas no formato DDD (Somente em Graus)

Latitude: -22.55109°

Longitude: -47.43001°

Execução no MATLAB:

```
>> [x,y,utmzone] = deg2utm(-22.55109,-47.43001);
```

Dados de Saída:

x = 2.5010e+05

y = 7.5041e+06

utmzone = 23 K

```

function [x,y,utmzone] =
deg2utm(Lat,Lon)

% Argument checking
%
error(nargchk(2, 2, nargin)); %2
arguments required
n1=length(Lat);
n2=length(Lon);
if (n1~=n2)
    error('Lat and Lon vectors should
have the same length');
end

% Memory pre-allocation
%
x=zeros(n1,1);
y=zeros(n1,1);
utmzone(n1,:)= '60 X';

% Main Loop
%
for i=1:n1
    la=Lat(i);
    lo=Lon(i);

    sa = 6378137.000000 ; sb =
6356752.314245;

    %e = ( ( ( sa ^ 2 ) - ( sb ^ 2 )
) ^ 0.5 ) / sa;
    e2 = ( ( ( sa ^ 2 ) - ( sb ^ 2 )
) ^ 0.5 ) / sb;
    e2cuadrada = e2 ^ 2;
    c = ( sa ^ 2 ) / sb;
    %alpha = ( sa - sb ) / sa;
%f
    %ablandamiento = 1 / alpha; %
1/f

    lat = la * ( pi / 180 );
    lon = lo * ( pi / 180 );

    Huso = fix( ( lo / 6 ) + 31);
    S = ( ( Huso * 6 ) - 183 );
    deltaS = lon - ( S * ( pi / 180 )
);

    a = cos(lat) * sin(deltaS);
    epsilon = 0.5 * log( ( 1 + a ) /
( 1 - a ) );
    nu = atan( tan(lat) / cos(deltaS)
) - lat;
    v = ( c / ( ( 1 + ( e2cuadrada *
( cos(lat) ) ^ 2 ) ) ) ^ 0.5 ) *
0.9996;
    ta = ( e2cuadrada / 2 ) * epsilon
^ 2 * ( cos(lat) ) ^ 2;
    a1 = sin( 2 * lat );
    a2 = a1 * ( cos(lat) ) ^ 2;
    j2 = lat + ( a1 / 2 );
    j4 = ( ( 3 * j2 ) + a2 ) / 4;
    j6 = ( ( 5 * j4 ) + ( a2 * (
cos(lat) ) ^ 2 ) ) / 3;
    alfa = ( 3 / 4 ) * e2cuadrada;
    beta = ( 5 / 3 ) * alfa ^ 2;
    gama = ( 35 / 27 ) * alfa ^ 3;
    Bm = 0.9996 * c * ( lat - alfa *
j2 + beta * j4 - gama * j6 );
    xx = epsilon * v * ( 1 + ( ta / 3
) ) + 500000;
    yy = nu * v * ( 1 + ta ) + Bm;

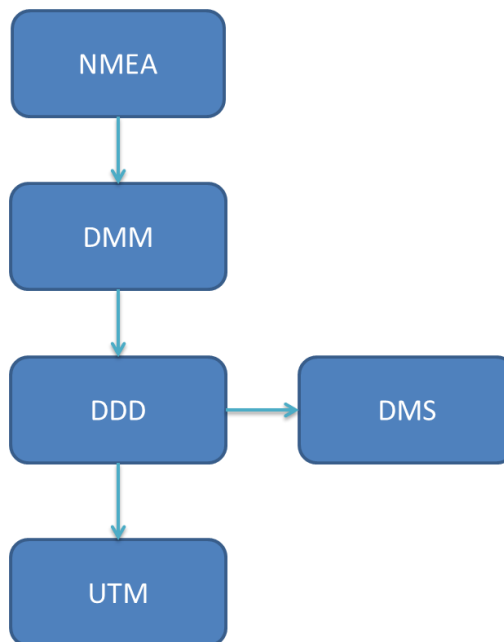
    if (yy<0)
        yy=9999999+yy;
    end

    x(i)=xx;
    y(i)=yy;
    utmzone(i,:)=sprintf('%02d
%c',Huso,Letra);
end

```

Quadro 5.2- Trecho do código MATLAB de conversão de coordenadas DDD para UTM. [23]

O código é responsável pela conversão do formato DDD (antes DMM) para UTM. O processo de extração dos dados NMEA até o UTM é melhor visualizado pela Figura 5.16.

Figura 5.16 - Processo de conversão por *software*.

Resumidamente, têm-se dados de GPS de Fase em coordenadas no formato UTM, este GPS de Fase e as coordenadas foram disponibilizados pelo professor e têm-se também as coordenadas de GPS de código para os percursos executados, lembrando que os dados de ambos dispositivos GPS foram coletados do mesmo atleta ao mesmo tempo. As coordenadas do GPS de código sofreram processos de conversão para a disponibilidade no formato UTM, assim sendo possível a comparação dos dados obtidos.

Por características do GPS de fase, a taxa de disponibilização de coordenadas é de 1 Hz, o GPS de código, porém, fornece na taxa de 5 Hz, portanto, nos resultados encontra-se maior número de pontos do GPS de código em relação ao de fase.

5.4.2 Resultados

Neste subcapítulo são disponibilizados os gráficos gerados pelo MATLAB dos dados do GPS de fase e de código, nos dois tipos de percursos e em duas velocidades repetido por uma vez, no total, oito gráficos. Em cada gráfico, a linha vermelha é o resultado da interpolação linear

das amostras do GPS de fase, cada amostra é representada por um quadrado vermelho. O GPS de código tem a linha azul, com as amostras representadas por cruces azuis.

Os Eixos X e Y de coordenadas UTM, foram normalizados para melhor comparação entre os resultados de todos os dispositivos GPS, as unidades dos eixos estão em metros.

Estão disponibilizados nas páginas seguintes os gráficos gerados pelo experimento.

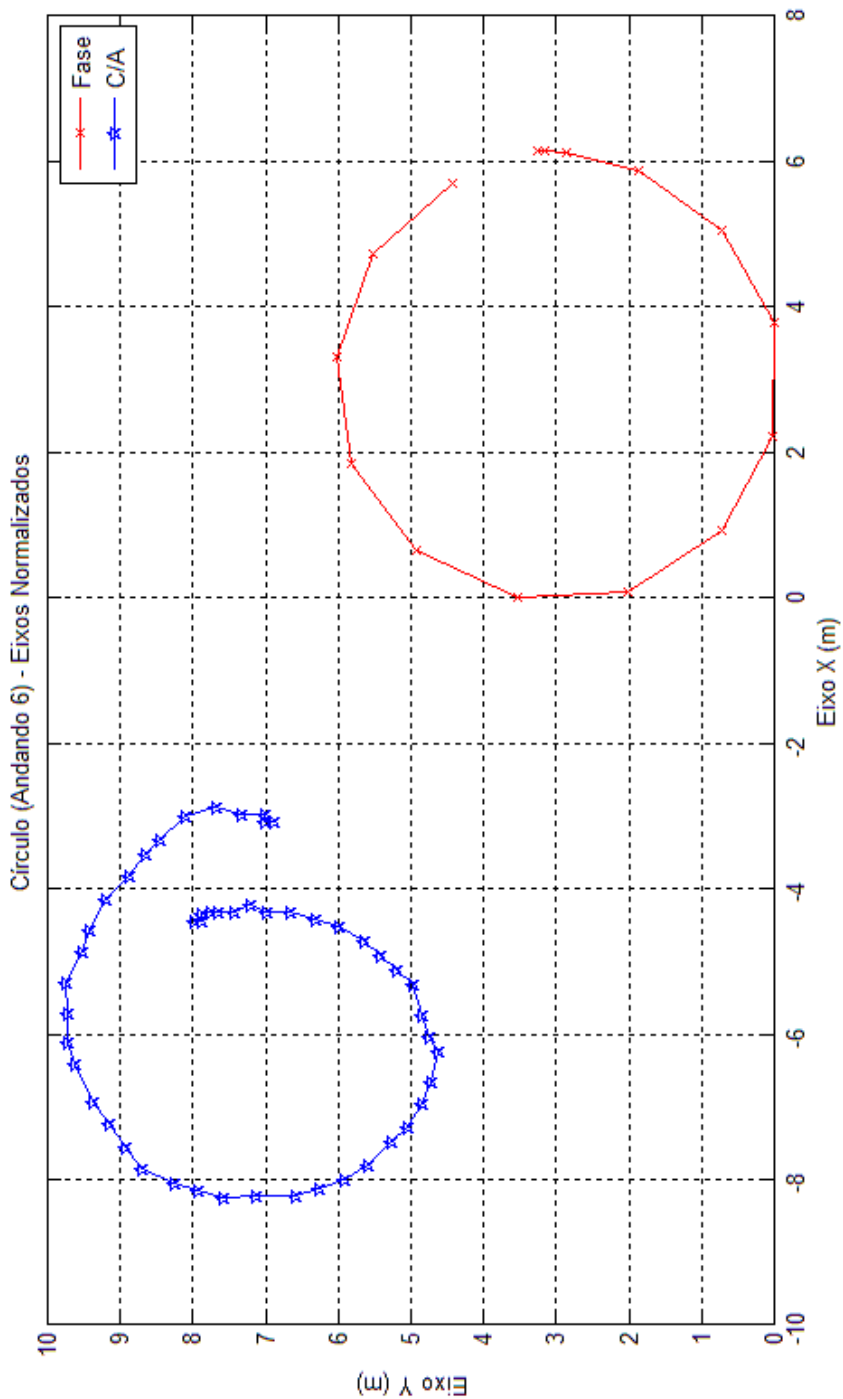


Figura 5.17 - Teste andando 6 (Círculo).

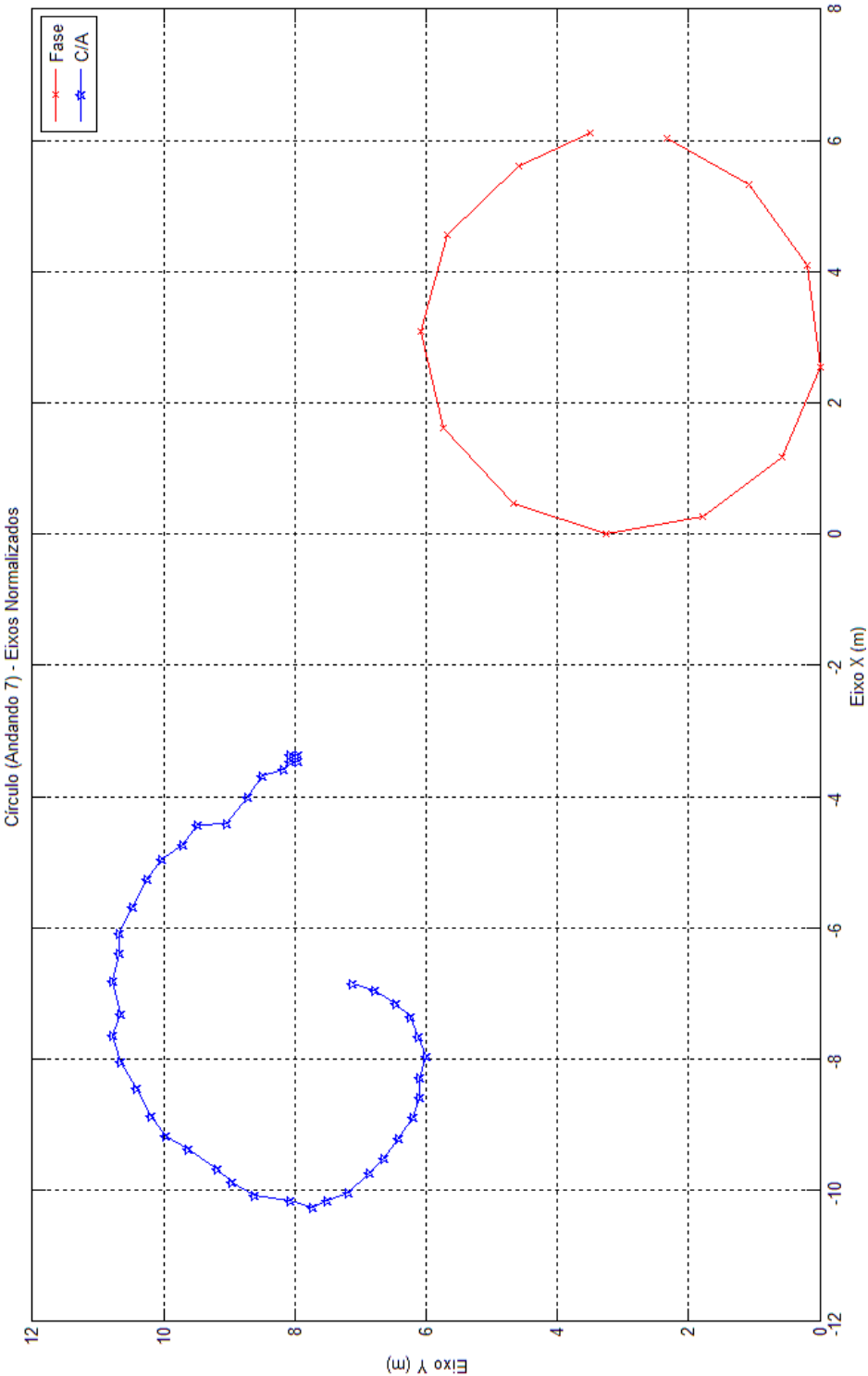


Figura 5.18 - Teste andando 7 (Círculo).

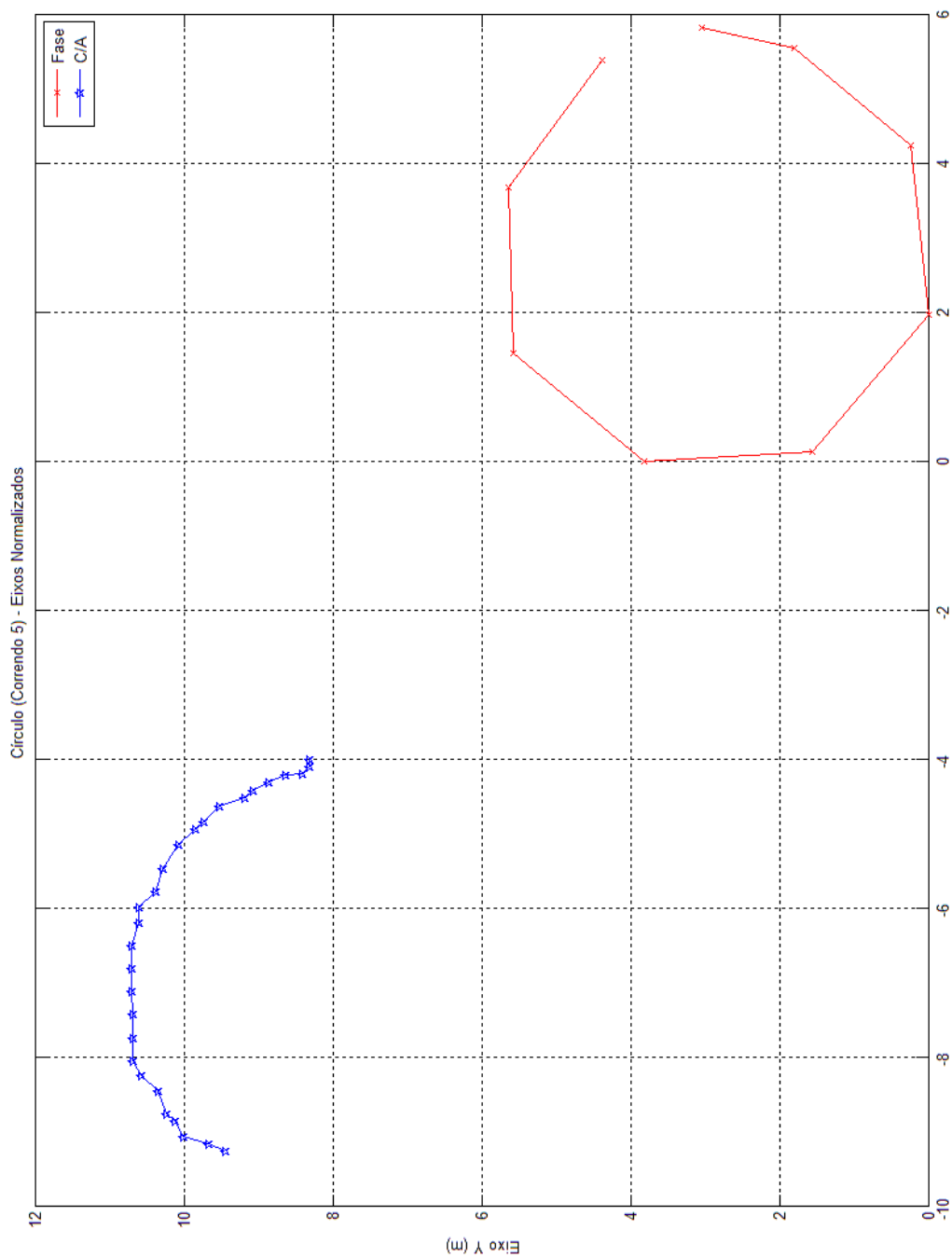


Figura 5.19 - Teste correndo 5 (Círculo).

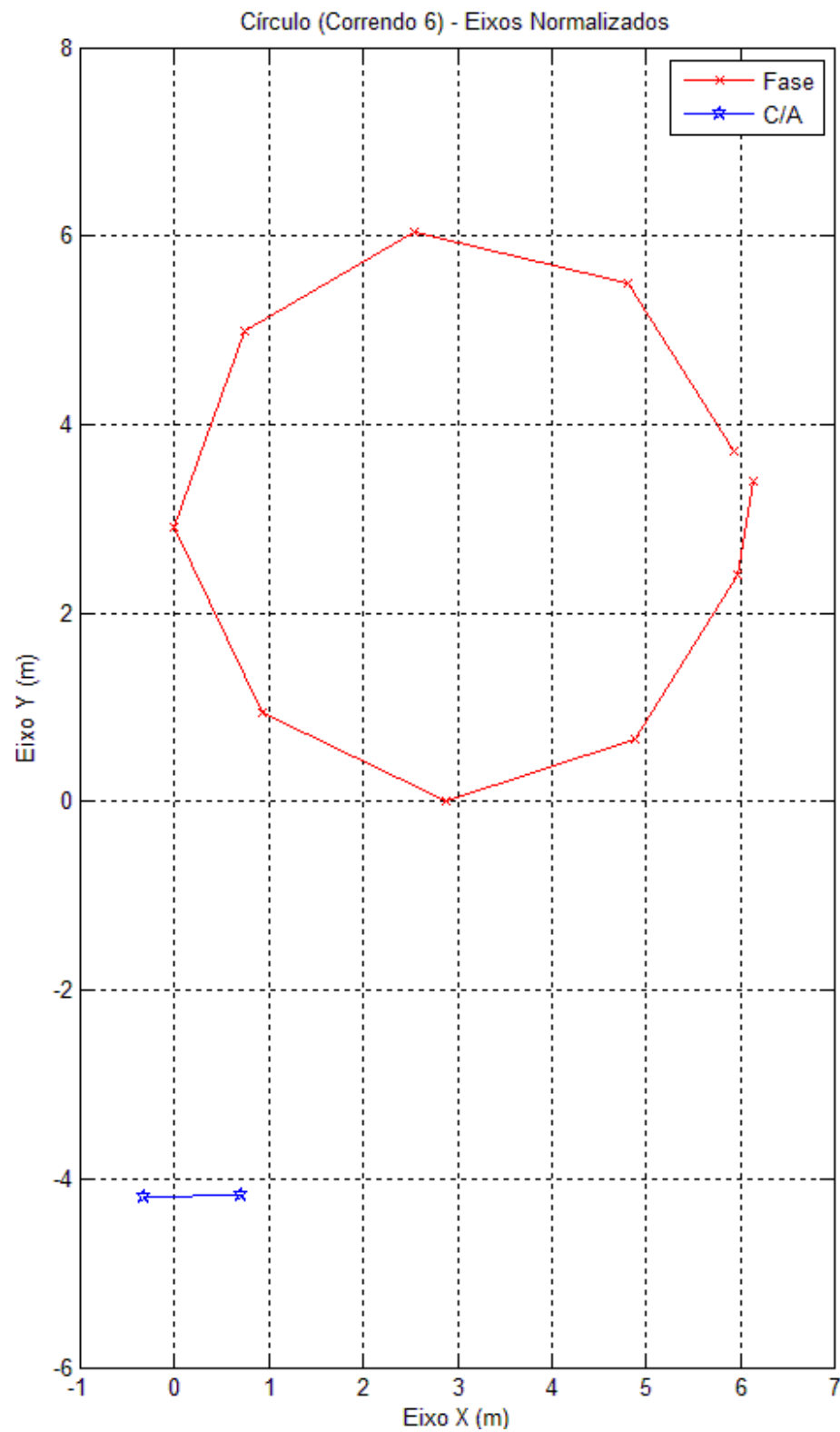


Figura 5.20 - Teste correndo 6 (Círculo).

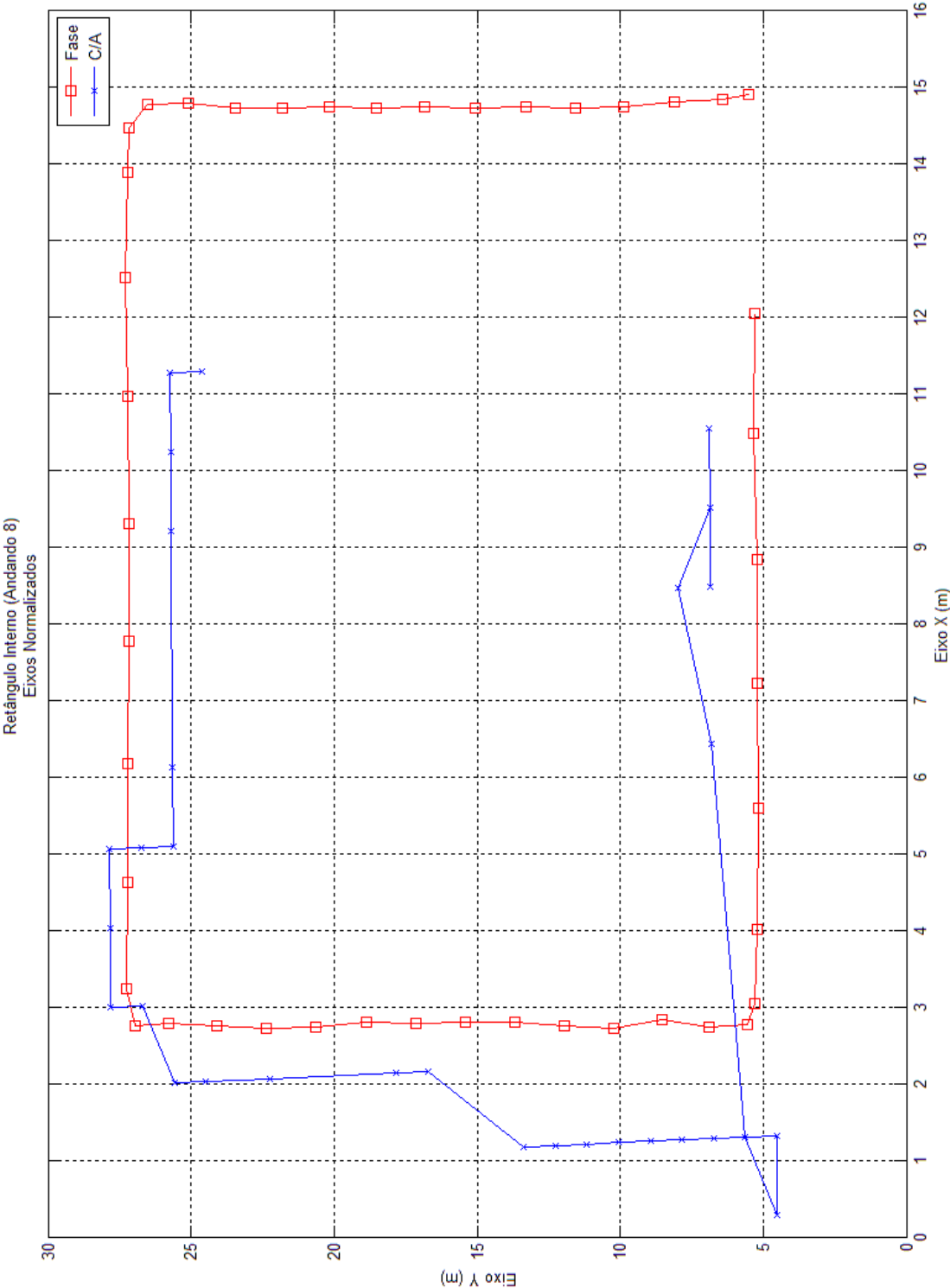


Figura 5.21 - Teste andando 8 (Retângulo Interno).

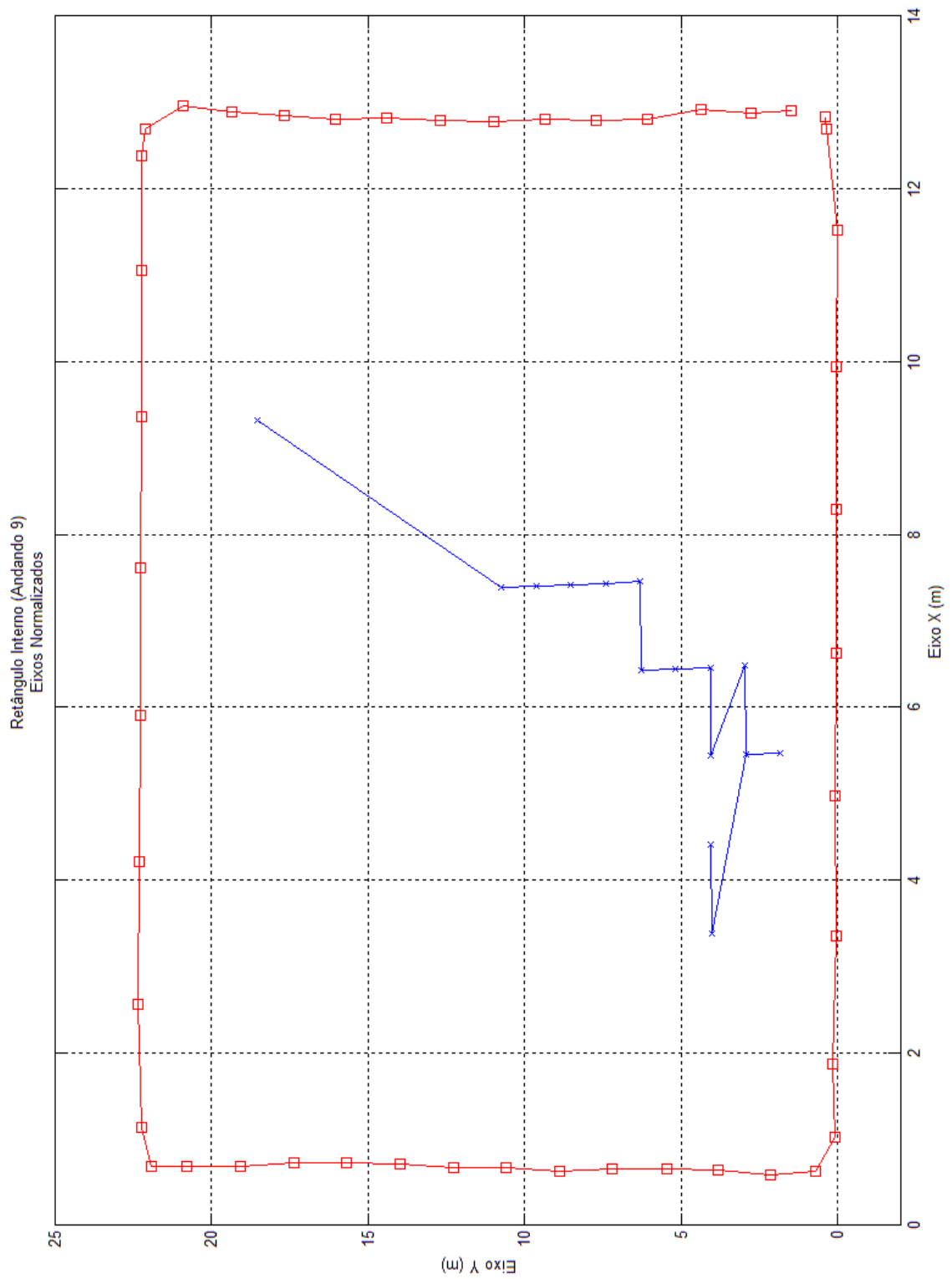


Figura 5.22 - Teste andando 9 (Retângulo Interno).

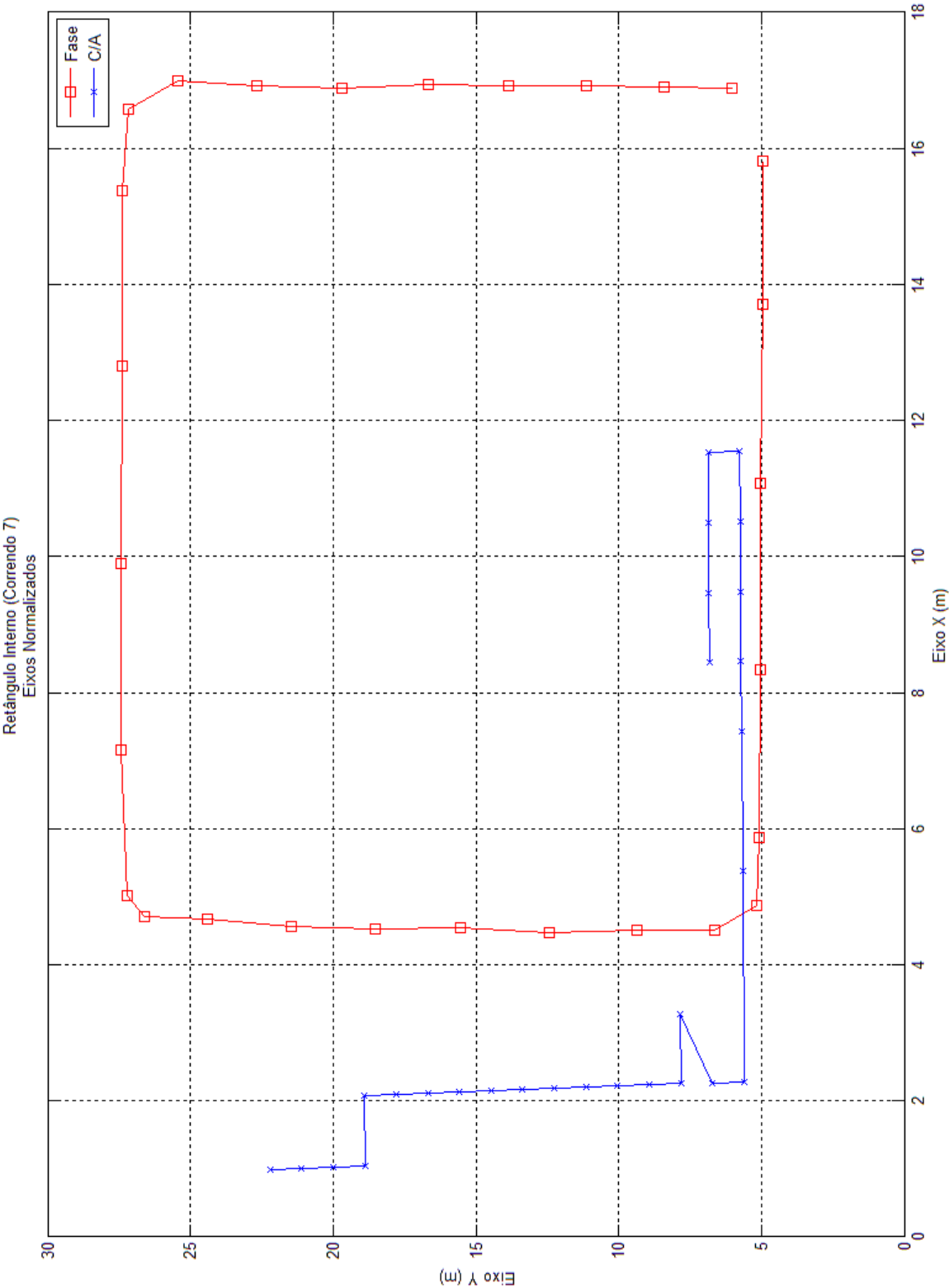


Figura 5.23 - Teste correndo 7 (Retângulo Interno).

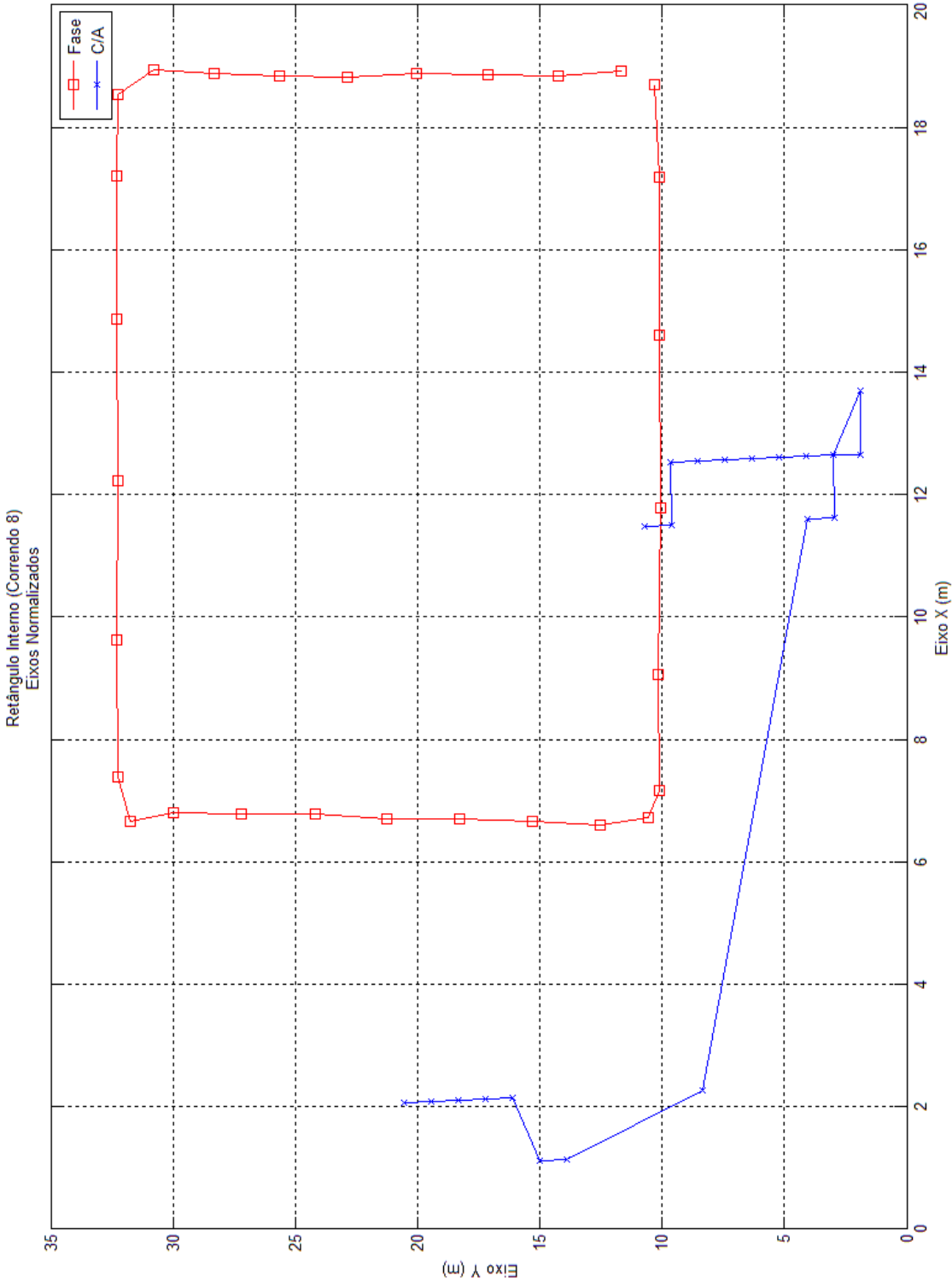


Figura 5.24 - Teste correndo 8 (Retângulo Interno).

6. Conclusões

Com a união do GPS e dispositivos ZigBee, o sistema proposto neste trabalho oferece uma gama de aplicações em relação aos tipos de modalidades esportivas em que ela pode ser empregada, não possuindo limitação de funcionamento em ambientes fechados e abertos.

Considerando os resultados dos testes, pode-se classificar a exatidão do sistema de trilateração com os módulos ZigBee, como razoável, atuando na faixa de um metro, porém, pode oferecer melhores resultados com um tratamento de alto nível dos dados de RSSI, como sugerido em [24], aonde se aplica uma filtragem dos valores através de padrões de qualidade do link sem fio, o fator LQI do ZigBee. Mesmo com a exatidão centrada na proximidade de metros, o sistema de trilateração pode funcionar em altíssima taxa de amostragem, ao contrário da grande maioria dos dispositivos GPS, que não passam de algumas amostras por segundo, além do fato de ser um sistema que é aplicado em ambientes fechados, fato que é praticamente inviável com o GPS.

O GPS de código empregado no sistema se comparado aos comerciais mais populares, tem boas características de funcionamento, mas mesmo assim, por ser de código, não é de se esperar padrões de posicionamento ou precisão mínima que seja menor que 22 metros para mais ou para menos. No caso dos 22 metros de precisão, uma aplicação *outdoor* numa quadra descoberta é inviável, pois supera as medidas de dimensão da quadra, porém no caso de uma maratona o uso do GPS de código é bem vindo.

Assim, com características até que grosseiras do GPS de código, o sistema ainda pode ser enquadrado em outras modalidades esportivas que não requerem muita precisão, mesmo caso para a trilateração com o ZigBee.

Sistemas de rastreamento comerciais custam na ordem de milhares de reais e são de propriedade fechada, não agregam muitos sensores e praticamente tem customização nula. O Sistema confeccionado neste trabalho utiliza tecnologias abertas, que sempre podem ser enriquecidas, agregando maiores funcionalidades, tem *hardware* simples que pode ser facilmente modificado para o acoplamento de novos sensores e é amigável a modificações. Consequentemente a um custo inferior aos sistemas comerciais.

Durante os testes aconteceram problemas frequentes com a duração das baterias de 9V, isso pois não possuem capacidade de carga elevada. Uma solução para isso é o uso de baterias de polímero, onde os problemas seriam minimizados. Porém essas baterias são caras e necessitam de

carregadores especiais que também são caros, e este fato foi considerado na composição do sistema.

6.1 Trabalhos Futuros

- Estudo de aperfeiçoamento dos métodos para estimação das distâncias e a possibilidade da predição com filtro de Kalman e utilização de filtros pelo fator de LQI [24].
- Composição do algoritmo em alto nível na estação gerenciadora para a disposição dos valores de posição dos dispositivos móveis e resultados dos sensores implantados nas plataformas, aumento da resolução dos valores de RSSI utilizando o recurso de *hardware* disponível pelo ZigBee, ou seja, construção de demodulador PWM de precisão.
- Uso de métodos inerciais para estimação das posições com o sensor MPU9250 como descrito por [25].
- Emprego de GPS de fase de alta taxa de aquisição exatidão como o Piksi [26].
- Implantação da tecnologia de posicionamento por ultrassom, tornando o SSR em um sistema híbrido de rastreamento (SHR).
- Alteração dos modos de funcionamento do ZigBee de AT para API, que possui mais recursos para o controle dos dados.
- Migração de microcontroladores de 8 bits para 32 bits.
- Avaliação comparativa experimental com outros sistemas comerciais e métodos para o rastreamento de objetos já existentes.
- Migração da tecnologia de confecção de PCB de SMT para SMD, para miniaturização dos dispositivos móveis.
- Aquisição de baterias de polímero de maior capacidade de carga.
- Modelagem e montagem de suportes anatômicos para acoplagem dos dispositivos nos atletas.
- Estudos com os padrões de irradiação das antenas e aperfeiçoamento da propagação dos sinais ZigBee, visando a melhora da diretividade.
- A busca de possíveis novas áreas de aplicações e contribuições do sistema.

7. Bibliografia

- [1] YU-TSO CHEN, C.-L. Y. Y.-K. C. C.-P. C. **A RSSI-based Algorithm for Indoor Localization. Using ZigBee in Wireless Sensor Network.** IEEE. [S.l.].
- [2] SATEL. Radio technology for RTK, 2013. Disponível em: <<http://satel.com>>. Acesso em: 2013.
- [3] LIMA JUNIOR, P. **O Laboratório de Mecânica.** IF-UFRGS. Porto Alegre. 2012.
- [4] ROBERT JARVIS, A. M. K. T. B. Z. **Indoor Positioning System.** EE 4820. [S.l.]. 2011.
- [5] MASASHI SUGANO, T. K. Y. O. M. M. **Indoor Localization System Using RSSI Measurement of Wireless Sensor Network Based on ZigBee Standard.** WSN 2006. [S.l.]. 2006.
- [6] GEO-POSITIONING, GPS, DGPS, and Positioning Accuracy. **Precision Agriculture and precision Farming,** 2009. Disponível em: <<http://www.precisionag.org/PDF/ch2.pdf>>. Acesso em: 2013.
- [7] MONICO, J. F. G. **Posicionamento pelo GNSS - Descrição, fundamentos e aplicações.** 2^a. ed. São Paulo: Unesp, 2008. ISBN: 978-85-7139-788-0.
- [8] OZIEL HERNANDEZ, V. J. S. C. P. B. **Position Location Monitoring Using IEEE® 802.15.4/ZigBee® technology.** IEEE. [S.l.].
- [9] MARKUS PICHLER, S. S. A. S. M. V. **Multi-Channel Distance Measurement With IEEE 802.15.4 (ZigBee) Devices.** IEEE. [S.l.].
- [10] INVENSENSE. MPU 9250 Datasheet. **Invensense,** 2014. Disponível em: <<http://www.invensense.com/mems/gyro/documents/PS-MPU-9250A-01.pdf>>.
- [11] LAB CENTER ELETRONICS. Isis 7 Professional. **Lab Center,** 2014. Disponível em: <<http://www.labcenter.com/index.cfm>>. Acesso em: 2014.
- [12] NATIONAL INSTRUMENTS. Ultiboard. **NI,** 2014. Disponível em: <<http://www.ni.com/ultiboard/>>. Acesso em: 2014.
- [13] CUSTOM COMPUTER SERVICES, INC. PCWHD IDE Compiler. **CCS Info,** 2014. Disponível em: <http://www.ccsinfo.com/product_info.php?products_id=PCWHD_full>.

Acesso em: 2014.

- [14] MICROCHIP TECHNOLOGY. MPLAB X IDE. **Microchip**, 2014. Disponível em: <<http://www.microchip.com/pagehandler/en-us/family/mplabx/>>. Acesso em: 2014.
- [15] VISUALGPS, LLC. Visual GPS Application. **Visual GPS**, 2013. Disponível em: <<http://www.visualgps.net/VisualGPS/>>. Acesso em: 2013.
- [16] DIGI INTERNATIONAL INC. X-CTU. **Digi**, 2013. Disponível em: <<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>>. Acesso em: 2013.
- [17] MATHWORKS. MATLAB R2013a. **Mathworks**, 2014. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em: 2014.
- [18] MESSIAS, A. R. Configurando os Módulos através de comandos ATs. **Rogercom**, 2014. Disponível em: <<http://www.rogercom.com/ZigBee/ZigBeePag03.htm>>. Acesso em: 2014.
- [19] THE LOCOSYS TECHNOLOGY INC. LS20031 Datasheet. **Sparkfun**, 2013. Disponível em: <http://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3_datasheet_v1.2.pdf>. Acesso em: 2013.
- [20] DIGI INTERNATIONAL INC. XBee® 802.15.4 Datasheet. **Digi**, 2014. Disponível em: <http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf>. Acesso em: 2014.
- [21] SYN NATSCHKE, S. Geographic Tools : Coordinate Conversion / Datum Transformation. **Secrets of the West**, 2014. Disponível em: <<http://www.synnatschke.de/geo-tools/coordinate-converter.php>>. Acesso em: 2014.
- [22] EMBARCADERO AND RAD STUDIO. Delphi 2010 IDE. **Embarcadero**, 2014. Disponível em: <<http://www.embarcadero.com/br/products/delphi>>. Acesso em: 2014.
- [23] PALACIOS, R. deg2utm. **MATLAB Central**, 2014. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/10915-deg2utm>>. Acesso em: 2014.
- [24] SHARLY JOANA HALDER, J. G. P. W. K. **Adaptive Filtering for Indoor Localization using ZIGBEE RSSI and LQI Measurement**. Yonsei University, Kyungpook National University. Seoul, Daegu. 2011.

[25] FOXLIN, E. **Pedestrian Tracking with Shoe-Mounted Inertial Sensors**. IEEE. [S.l.]. 2005.

[26] PIKSI: The RTK GPS Receiver by Swift Navigation Inc. **Kickstarter**, 2013. Disponível em: <<https://www.kickstarter.com/projects/swiftnav/piksi-the-rtk-gps-receiver>>. Acesso em: 2013.

8. Anexo e Apêndice

Função deg2utm para MATLAB

```
function [x,y,utmzone] = deg2utm(Lat,Lon)
% -----
% [x,y,utmzone] = deg2utm(Lat,Lon)
%
% Description: Function to convert lat/lon vectors into UTM coordinates (WGS84).
% Some code has been extracted from UTM.m function by Gabriel Ruiz Martinez.
%
% Inputs:
%   Lat: Latitude vector.   Degrees.   +ddd.ddddd   WGS84
%   Lon: Longitude vector.  Degrees.   +ddd.ddddd   WGS84
%
% Outputs:
%   x, y , utmzone.   See example
%
% Example 1:
%   Lat=[40.3154333; 46.283900; 37.577833; 28.645650; 38.855550; 25.061783];
%   Lon=[-3.4857166; 7.8012333; -119.95525; -17.759533; -94.7990166; 121.640266];
%   [x,y,utmzone] = deg2utm(Lat,Lon);
%   fprintf('%7.0f ',x)
%       458731  407653  239027  230253  343898  362850
%   fprintf('%7.0f ',y)
%       4462881  5126290  4163083  3171843  4302285  2772478
%   utmzone =
%       30 T
%       32 T
%       11 S
%       28 R
%       15 S
%       51 R
%
% Example 2: If you have Lat/Lon coordinates in Degrees, Minutes and Seconds
%   LatDMS=[40 18 55.56; 46 17 2.04];
%   LonDMS=[-3 29 8.58; 7 48 4.44];
%   Lat=dms2deg(mat2dms(LatDMS)); %convert into degrees
%   Lon=dms2deg(mat2dms(LonDMS)); %convert into degrees
%   [x,y,utmzone] = deg2utm(Lat,Lon)
%
% Author:
%   Rafael Palacios
%   Universidad Pontificia Comillas
%   Madrid, Spain
% Version: Apr/06, Jun/06, Aug/06, Aug/06
% Aug/06: fixed a problem (found by Rodolphe Dewarret) related to southern
%   hemisphere coordinates.
% Aug/06: corrected m-Lint warnings
```

```

%-----
% Argument checking
%
error(nargchk(2, 2, nargin)); %2 arguments required
n1=length(Lat);
n2=length(Lon);
if (n1~=n2)
    error('Lat and Lon vectors should have the same length');
end

% Memory pre-allocation
%
x=zeros(n1,1);
y=zeros(n1,1);
utmzone(n1,:)= '60 X';

% Main Loop
%
for i=1:n1
    la=Lat(i);
    lo=Lon(i);

    sa = 6378137.000000 ; sb = 6356752.314245;

    %e = ( ( ( sa ^ 2 ) - ( sb ^ 2 ) ) ^ 0.5 ) / sa;
    e2 = ( ( ( sa ^ 2 ) - ( sb ^ 2 ) ) ^ 0.5 ) / sb;
    e2cuadrada = e2 ^ 2;
    c = ( sa ^ 2 ) / sb;
    %alpha = ( sa - sb ) / sa; %f
    %ablandamiento = 1 / alpha; % 1/f

    lat = la * ( pi / 180 );
    lon = lo * ( pi / 180 );

    Huso = fix( ( lo / 6 ) + 31);
    S = ( ( Huso * 6 ) - 183 );
    deltaS = lon - ( S * ( pi / 180 ) );

    if (la<-72), Letra='C';
    elseif (la<-64), Letra='D';
    elseif (la<-56), Letra='E';
    elseif (la<-48), Letra='F';
    elseif (la<-40), Letra='G';
    elseif (la<-32), Letra='H';
    elseif (la<-24), Letra='J';
    elseif (la<-16), Letra='K';
    elseif (la<-8), Letra='L';
    elseif (la<0), Letra='M';
    elseif (la<8), Letra='N';
    elseif (la<16), Letra='P';
    elseif (la<24), Letra='Q';
    elseif (la<32), Letra='R';
    elseif (la<40), Letra='S';
    elseif (la<48), Letra='T';
    elseif (la<56), Letra='U';
    elseif (la<64), Letra='V';
    elseif (la<72), Letra='W';
    else Letra='X';
    end

    a = cos(lat) * sin(deltaS);
    epsilon = 0.5 * log( ( 1 + a ) / ( 1 - a ) );
    nu = atan( tan(lat) / cos(deltaS) ) - lat;
    v = ( c / ( ( 1 + ( e2cuadrada * ( cos(lat) ) ^ 2 ) ) ) ^ 0.5 ) * 0.9996;
    ta = ( e2cuadrada / 2 ) * epsilon ^ 2 * ( cos(lat) ) ^ 2;
    a1 = sin( 2 * lat );
    a2 = a1 * ( cos(lat) ) ^ 2;
    j2 = lat + ( a1 / 2 );

```



```

j4 = ( ( 3 * j2 ) + a2 ) / 4;
j6 = ( ( 5 * j4 ) + ( a2 * ( cos(lat) ) ^ 2 ) ) / 3;
alfa = ( 3 / 4 ) * e2cuadrada;
beta = ( 5 / 3 ) * alfa ^ 2;
gama = ( 35 / 27 ) * alfa ^ 3;
Bm = 0.9996 * c * ( lat - alfa * j2 + beta * j4 - gama * j6 );
xx = epsilon * v * ( 1 + ( ta / 3 ) ) + 500000;
yy = nu * v * ( 1 + ta ) + Bm;

if (yy<0)
    yy=9999999+yy;
end

x(i)=xx;
y(i)=yy;
utmzone(i,:)=sprintf('%02d %c',Huso,Letra);
end

```

Código do *firmware* das estações móveis nos testes de GPS

```

#include <18F4550.h>
#define HIGH_INTS=TRUE
#define fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDIV1, VREGEN, NOXINST
#define use delay(clock=48M)

#define use rs232(baud=9600, UART1, STREAM=ZIGBEE)
#define use rs232(baud=9600, xmit=PIN_D0, rcv=PIN_B2, STREAM=GPS, ERRORS, SAMPLE_EARLY)

#define int_rda
void zigbee_isr()
{
    char a;

    if(kbhit())
    {
        a = fgetc(ZIGBEE);
        fputc(a,GPS);
    }
}

#define int_ext2 HIGH
void gps_isr()
{
    char a;

    if(kbhit(GPS))
    {
        a = fgetc(GPS);
        fputc(a,ZIGBEE);
    }
}

void main()
{
    enable_interrupts(INT_EXT2);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    ext_int_edge( 2, H_TO_L);

    while(true)
    {
        output_high(PIN_D2);
        delay_ms(980);
        output_low(PIN_D2);
        delay_ms(20);
    }
}

```

```
}

```

Código fonte do programa conversor de coordenadas.

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, strutils, ExtCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    ListBox1: TListBox;
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label5: TLabel;
    Label6: TLabel;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Label7: TLabel;
    OpenFileDialog1: TOpenDialog;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Edit10: TEdit;
    Edit11: TEdit;
    Edit12: TEdit;
    Label8: TLabel;
    Label9: TLabel;
    Edit13: TEdit;
    Edit14: TEdit;
    Edit15: TEdit;
    Edit16: TEdit;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Bevel1: TBevel;
    Label10: TLabel;
    SaveDialog1: TSaveDialog;
    Label11: TLabel;
    Edit17: TEdit;
    procedure Button1Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
```

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  if OpenFileDialog1.Execute then
  begin
    Listbox1.Items.LoadFromFile(OpenDialog1.FileName);
  end;
end;

procedure TForm1.Button2Click(Sender: TObject); //DMS
var
  dd, d, m, s : real;
  i : integer;
  txt : textfile;
  str : string;
begin
  if SaveDialog1.Execute then
  begin
    AssignFile(txt, SaveDialog1.FileName);
    Rewrite(txt);

    for i := 0 to Listbox1.Items.Count - 1 do
    begin
      edit17.Text := copy(Listbox1.Items[i], 8, 10);

      edit17.Text := copy(edit17.Text, 1, 2) + ':' + copy(edit17.Text, 3, 2) + ':' +
        copy(edit17.Text, 5, 2) + ',' + copy(edit17.Text, 8, 3);

      //LATITUDE

      //DMM
      edit4.Text := copy(Listbox1.Items[i], 19, 2); //ddd
      edit5.Text := copy(Listbox1.Items[i], 21, 7); //mm.mmmmm
      edit5.Text := AnsiReplaceStr(edit5.Text, '.', ',');

      edit14.Text := copy(Listbox1.Items[i], 29, 1);
      edit16.Text := edit14.Text;

      //DDD
      edit6.text := floattostr(strtoint(edit4.Text) + (strtofloat(edit5.Text)/60));
      edit6.Text := FormatFloat('00.00000', StrtoFloat(edit6.Text));

      //DMS
      dd := strtofloat(edit6.Text);
      d := trunc(dd);
      m := trunc((dd-d)*60);
      s := (dd - d - m/60) * 3600;

      edit1.Text := floattostr(d);
      edit2.Text := floattostr(m);
      edit3.Text := floattostr(s);
      edit3.Text := FormatFloat('00.00000', StrtoFloat(edit3.Text));

      if edit14.Text = 'S' then edit6.text := '-' + edit6.text;

      //LONGITUDE

      //DMM
      edit10.Text := copy(Listbox1.Items[i], 31, 3); //ddd
      edit11.Text := copy(Listbox1.Items[i], 34, 7); //mm.mmmmm
      edit11.Text := AnsiReplaceStr(edit11.Text, '.', ',');

      edit13.Text := copy(Listbox1.Items[i], 42, 1);
      edit13.Text := AnsiReplaceStr(edit13.Text, 'W', 'O');
      edit13.Text := AnsiReplaceStr(edit13.Text, 'E', 'L');
      edit15.Text := edit13.Text;

      //DDD
      edit12.text := floattostr(strtoint(edit10.Text) + (strtofloat(edit11.Text)/60));

```

```

edit12.Text := FormatFloat('00.00000', StrtoFloat(edit12.Text));

//DMS
dd := strtofloat(edit12.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit7.Text := floattostr(d);
edit8.Text := floattostr(m);
edit9.Text := floattostr(s);
edit9.Text := FormatFloat('00.00000', StrtoFloat(edit9.Text));

if edit13.Text = 'O' then edit12.text := '-' + edit12.text;

str := edit17.text+';'+
edit1.Text +'° ' + edit2.text +''' ' + edit3.text + '''' ' + edit16.text+';'+
edit7.Text +'° ' + edit8.text +''' ' + edit9.text + '''' ' + edit15.text;
Writeln(txt,str);
end;
CloseFile(txt);
end;
end;

procedure TForm1.Button3Click(Sender: TObject);
var
dd, d, m, s : real;
i : integer;
txt : textfile;
str : string;
begin

if SaveDialog1.Execute then
begin
AssignFile(txt,SaveDialog1.FileName);
Rewrite(txt);

for i := 0 to ListBox1.Items.Count - 1 do
begin
edit17.Text := copy(ListBox1.Items[i],8,10);

edit17.Text := copy(edit17.Text,1,2)+':'+copy(edit17.Text,3,2)+':'+
copy(edit17.Text,5,2)+','+copy(edit17.Text,8,3);

//LATITUDE

//DMM
edit4.Text := copy(ListBox1.Items[i],19,2); //ddd
edit5.Text := copy(ListBox1.Items[i],21,7); //mm.mmmmm
edit5.Text := AnsiReplaceStr(edit5.Text,',','');

edit14.Text := copy(ListBox1.Items[i],29,1);
edit16.Text := edit14.Text;

//DDD
edit6.text := floattostr(strtoint(edit4.Text) + (strtfloat(edit5.Text)/60));
edit6.Text := FormatFloat('00.00000', StrtoFloat(edit6.Text));

//DMS
dd := strtfloat(edit6.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit1.Text := floattostr(d);
edit2.Text := floattostr(m);
edit3.Text := floattostr(s);
edit3.Text := FormatFloat('00.00000', StrtoFloat(edit3.Text));

if edit14.Text = 'S' then edit6.text := '-' + edit6.text;

```

```

//LONGITUDE

//DMM
edit10.Text := copy(ListBox1.Items[i],31,3); //ddd
edit11.Text := copy(ListBox1.Items[i],34,7); //mm.mmmmm
edit11.Text := AnsiReplaceStr(edit11.Text, '.', ',');

edit13.Text := copy(ListBox1.Items[i],42,1);
edit13.Text := AnsiReplaceStr(edit13.Text, 'W', 'O');
edit13.Text := AnsiReplaceStr(edit13.Text, 'E', 'L');
edit15.Text := edit13.Text;

//DDD
edit12.text := floattostr(strtoint(edit10.Text) + (strtofloat(edit11.Text)/60));
edit12.Text := FormatFloat('00.00000', StrtoFloat(edit12.Text));

//DMS
dd := strtofloat(edit12.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit7.Text := floattostr(d);
edit8.Text := floattostr(m);
edit9.Text := floattostr(s);
edit9.Text := FormatFloat('00.00000', StrtoFloat(edit9.Text));

if edit13.Text = 'O' then edit12.text := '-' + edit12.text;

str := edit17.text+';'+
edit4.Text +'° '+ edit5.text +''' '+ edit14.text+';'+
edit10.Text +'° '+ edit11.text +''' '+ edit13.text;
Writeln(txt,str);
end;
CloseFile(txt);
end;
end;

procedure TForm1.Button4Click(Sender: TObject); //Exportar DDD
var
dd, d, m, s : real;
i : integer;
txt : textfile;
str : string;
begin

if SaveDialog1.Execute then
begin
AssignFile(txt,SaveDialog1.FileName);
Rewrite(txt);

for i := 0 to ListBox1.Items.Count - 1 do
begin
edit17.Text := copy(ListBox1.Items[i],8,10);

edit17.Text := copy(edit17.Text,1,2)+':'+copy(edit17.Text,3,2)+':'+
copy(edit17.Text,5,2)+', '+copy(edit17.Text,8,3);

//LATITUDE

//DMM
edit4.Text := copy(ListBox1.Items[i],19,2); //ddd
edit5.Text := copy(ListBox1.Items[i],21,7); //mm.mmmmm
edit5.Text := AnsiReplaceStr(edit5.Text, '.', ',');

edit14.Text := copy(ListBox1.Items[i],29,1);
edit16.Text := edit14.Text;

```

```

//DDD
edit6.text := floattostr(strtoint(edit4.Text) + (strtofloat(edit5.Text)/60));
edit6.Text := FormatFloat('00.00000', StrtoFloat(edit6.Text));

//DMS
dd := strtofloat(edit6.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit1.Text := floattostr(d);
edit2.Text := floattostr(m);
edit3.Text := floattostr(s);
edit3.Text := FormatFloat('00.00000', StrtoFloat(edit3.Text));

if edit14.Text = 'S' then edit6.text := '-' + edit6.text;

//LONGITUDE

//DMM
edit10.Text := copy(ListBox1.Items[i],31,3); //ddd
edit11.Text := copy(ListBox1.Items[i],34,7); //mm.mmmm
edit11.Text := AnsiReplaceStr(edit11.Text, '.', ',');

edit13.Text := copy(ListBox1.Items[i],42,1);
edit13.Text := AnsiReplaceStr(edit13.Text, 'W', 'O');
edit13.Text := AnsiReplaceStr(edit13.Text, 'E', 'L');
edit15.Text := edit13.Text;

//DDD
edit12.text := floattostr(strtoint(edit10.Text) + (strtofloat(edit11.Text)/60));
edit12.Text := FormatFloat('00.00000', StrtoFloat(edit12.Text));

//DMS
dd := strtofloat(edit12.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit7.Text := floattostr(d);
edit8.Text := floattostr(m);
edit9.Text := floattostr(s);
edit9.Text := FormatFloat('00.00000', StrtoFloat(edit9.Text));

if edit13.Text = 'O' then edit12.text := '-' + edit12.text;

str := edit17.Text + ';' + edit6.text + ';' + edit12.text;
Writeln(txt,str);
end;
CloseFile(txt);
end;
end;

procedure TForm1.ListBox1Click(Sender: TObject);
var
dd, d, m, s : real;
begin
edit17.Text := copy(ListBox1.Items[Listbox1.ItemIndex],8,10);

edit17.Text := copy(edit17.Text,1,2)+':'+copy(edit17.Text,3,2)+':'+
copy(edit17.Text,5,2)+','+copy(edit17.Text,8,3);

//LATITUDE

//DMM
edit4.Text := copy(ListBox1.Items[Listbox1.ItemIndex],19,2); //ddd
edit5.Text := copy(ListBox1.Items[Listbox1.ItemIndex],21,7); //mm.mmmm
edit5.Text := AnsiReplaceStr(edit5.Text, '.', ',');

edit14.Text := copy(ListBox1.Items[Listbox1.ItemIndex],29,1);

```

```

edit16.Text := edit14.Text;

//DDD
edit6.text := floattostr(strtoint(edit4.Text) + (strtofloat(edit5.Text)/60));
edit6.Text := FormatFloat('00.00000000', StrtoFloat(edit6.Text));

//DMS
dd := strtofloat(edit6.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit1.Text := floattostr(d);
edit2.Text := floattostr(m);
edit3.Text := floattostr(s);
edit3.Text := FormatFloat('00.00000', StrtoFloat(edit3.Text));

if edit14.Text = 'S' then edit6.text := '-' + edit6.text;

//LONGITUDE

//DMM
edit10.Text := copy(ListBox1.Items[Listbox1.ItemIndex],31,3); //ddd
edit11.Text := copy(ListBox1.Items[Listbox1.ItemIndex],34,7); //mm.mmmm
edit11.Text := AnsiReplaceStr(edit11.Text, '.', ',');

edit13.Text := copy(ListBox1.Items[Listbox1.ItemIndex],42,1);
edit13.Text := AnsiReplaceStr(edit13.Text, 'W', 'O');
edit13.Text := AnsiReplaceStr(edit13.Text, 'E', 'L');
edit15.Text := edit13.Text;

//DDD
edit12.text := floattostr(strtoint(edit10.Text) + (strtofloat(edit11.Text)/60));
edit12.Text := FormatFloat('00.00000', StrtoFloat(edit12.Text));

//DMS
dd := strtofloat(edit12.Text);
d := trunc(dd);
m := trunc((dd-d)*60);
s := (dd - d - m/60) * 3600;

edit7.Text := floattostr(d);
edit8.Text := floattostr(m);
edit9.Text := floattostr(s);
edit9.Text := FormatFloat('00.00000', StrtoFloat(edit9.Text));

if edit13.Text = 'O' then edit12.text := '-' + edit12.text;
end;

end.

```